Discrete Mathematics

Joseph Erickson

Table of Contents

1
Logic

1.1 – Statements

We take the concept of a **sentence** to be understood, and define the four principal types of sentences.

Definition 1.1. *A sentence that makes a declaration or assertion is* ***declarative****. A sentence that asks a questions is* ***interrogative****. A sentence that gives a command or instructions is* ***imperative****. A sentence that makes an emotional exclamation is* ***exclamatory****.*

A **statement** is a declarative sentence that may be judged to be true or false, but not both. The terms **true** and **false** are called **truth values**, and **symbolic logic**, which is the study of logic using symbols rather than words, denotes the value "true" by either the symbol T or 1, and the value "false" by F or 0. Certainly a question such as "What time is it?" cannot be meaningfully assigned a truth value, nor can a command such as "Go forth and conquer" or an exclamation of surprise such as "What the deuce!"

An exclamatory sentence such as "What a horrible ordeal!" may appear to be declarative, and perhaps even be a statement, but the simple fact is that what is being expressed by such a sentence is strictly a matter of opinion. One hour in a crawling traffic jam may seem a little thing to a pioneer who crossed the North American continent in a covered wagon.

Example 1.2. The sentence "A gluon mediates the strong force interaction between quarks" is declarative, and it is also a statement since it is true according to the Standard Model of particle physics.

The sentence "Millicent sold some pomegranates to what's-his-name" is declarative, but it is not a statement since the truth or falseness of the sentence hinges on who, precisely, is "what's-his-name."

The sentence "He told me he understood quantum field theory" is declarative, but it is not a statement since the truth or falseness of the sentence depends on to whom the pronoun "he" is referring.

The sentence "This sentence is false" is declarative, but it is not a statement since assigning it either truth value leads to a contradiction: If the sentence is taken to be true, then it must be false; but if it is taken to be false, then it must be true. This is a version of what is called the **liar's paradox**. ■

Example 1.3. The equation $2x + 6 = 0$ represents the declarative sentence "$2x + 6$ is equal to 0." It is not a statement since the truth or falseness of the sentence depends on what is substituted for the variable (as the term is used in algebra) denoted by x. Indeed, the sentence is true only when $x = -3$, otherwise it is false. ■

An **open sentence** is a declarative sentence whose truth or falseness depends on the values of one or more variables contained in the sentence.

3

Methods of Proof

3.1 – Logical Quantifiers

In Chapter 1 we began to develop a system of symbolic logic, but stopped short of its completion in order to spend Chapter 2 becoming acquainted with aspects of set theory. Now set theory will facilitate our further development of symbolic logic to include statements featuring terms like *for all, there exists, some, none*, and so on. Such terms indicate the presence of what is called a **logical quantifier**.

Definition 3.1. *Let S be a set, and let $P(x)$ be an open sentence over domain S. The **universal quantification** of $P(x)$ is "$P(x)$ for all x in S," which is written as $\forall x \in S\ P(x)$. The symbol $\forall$ denotes the **universal quantifier**, which is read as "for all."*

The expression $\forall x \in S\ P(x)$ is often read as "For all $x \in S$, $P(x)$," with the word "all" readily replaced by "every" or "each." We may even write $\forall x \in S\ P(x)$ as an implication: "If $x \in S$, then $P(x)$." We consider $\forall x \in S\ P(x)$ to be a true statement only if $P(x)$ is true for *every* $x \in S$.

Example 3.2. For an odd integer n, define the open sentence $P(n) : 7n + 4$ is odd. Letting S be the set of odd integers, we may certainly express the statement $\forall n \in S\ P(n)$ as "For all $n \in S\ P(n)$," but in plainer English we may also write

For every odd integer n, $7n + 4$ is odd,

or

If n is an odd integer, then $7n + 4$ is odd.

Other variations are possible. ■

Definition 3.3. *Let S be a set, and let $P(x)$ be an open sentence over domain S. The **existential quantification** of $P(x)$ is "There exists some x in S such that $P(x)$," which is written as $\exists x \in S\ P(x)$. The symbol $\exists$ denotes the **existential quantifier**, which is read as "there exists."*

The expression $\exists x \in S\ P(x)$ is often read as "There is some $x \in S$ such that $P(x)$," or "For some $x \in S$, $P(x)$," or "For at least one $x \in S$, $P(x)$." We consider $\exists x \in S\ P(x)$ to be a true statement only if $P(x)$ is true for *at least one* $x \in S$.

Any statement involving at least one quantifier is known as a **quantified statement**. A quantified statement of the form $\forall x \in S(\square)$, where either an open sentence or another quantified statement may be substituted for $\square$, is called a **universal statement**, while $\exists x \in S(\square)$ is an **existential statement**. Whatever is substituted for $\square$ in either $\forall x \in S(\square)$ or $\exists x \in S(\square)$ is said to lie in the **scope** of the quantifier written to the left of $\square$.

Example 3.4. Let A be the set of all animals, and define the open sentences

$$F(x):\ x \text{ is furry}$$
$$H(x):\ x \text{ hops}$$

We may translate the quantified statement $\exists x \in A(F(x) \wedge H(x))$ in the following ways, in order of increasingly plain English:

There exists an animal x such that $F(x)$ and $H(x)$.

There is an animal x such that x is furry and x hops.

There is an animal that is furry and hops.

Some animal is furry and hops.

To be sure, other variations are possible. ■

Definition 3.5. *Let Q_1 and Q_2 be two quantified statements involving a single variable x. If the truth values of Q_1 and Q_2 are the same for any choice of domain S for x, then Q_1 and Q_2 are said to be* ***logically equivalent*** *(or simply* ***equivalent****), and we write $Q_1 \equiv Q_2$.*

The negation of a quantified statement arises often in mathematical inquiries, and the following theorem gives two equivalencies involving such an operation.

Theorem 3.6. *For any open sentence $P(x)$ over domain S,*
1. $\neg(\forall x \in S\ P(x)) \equiv \exists x \in S\ (\neg P(x))$.
2. $\neg(\exists x \in S\ P(x)) \equiv \forall x \in S\ (\neg P(x))$.

Proof.
Proof of (1). To state $\neg(\forall x \in S\ P(x))$ means to say "It is not the case that $\forall x \in S\ P(x)$," or "It is not the case that $P(x)$ for every $x \in S$." Thus there must exist at least one $x \in S$ for which $P(x)$ is not the case, which we may state as: "There exists $x \in S$ such that $\neg P(x)$." This is precisely what $\exists x \in S\ (\neg P(x))$ states, and so the quantified statements $\neg(\forall x \in S\ P(x))$ and $\exists x \in S\ (\neg P(x))$ state the same thing.

Proof of (2). We could accomplish the proof in the same manner as in part (1), but will instead illustrate a different strategy. Substitute $\neg P(x)$ for $P(x)$ in the equivalency in part (1) to obtain

$$\neg(\forall x \in S\ (\neg P(x))) \equiv \exists x \in S\ (\neg(\neg P(x))).$$

We recall that $\neg(\neg P) \equiv P$ in general, and so obtain

$$\neg(\forall x \in S\ (\neg P(x))) \equiv \exists x \in S\ P(x).$$

Negating the statements on both sides of this equivalency yields

$$\neg(\neg(\forall x \in S\ (\neg P(x)))) \equiv \neg(\exists x \in S\ P(x)).$$

Once again we employ the equivalency $\neg(\neg P) \equiv P$ to arrive at

$$\forall x \in S\ (\neg P(x)) \equiv \neg(\exists x \in S\ P(x)). \tag{3.1}$$

Of course, $Q_1 \equiv Q_2$ if and only if $Q_2 \equiv Q_1$, and so from (3.1) we readily obtain the second equivalency stated in the theorem. ■

Example 3.7. Using an existential quantifier, state the negation of the statement "For every rational number r, the number $1/r$ is rational."

Solution. Certainly one way the negation of the given statement may be written is

It is not the case that, for every rational number r, the number $1/r$ is rational,

but this does not feature an existential quantifier. With part (1) of Theorem 3.6 we may write the negation as

There exists a rational number r such that the number $1/r$ is not rational.

Substituting "is" for "exists" is an acceptable alternative. ■

A quantified statement frequently features more than one quantifier. For instance, given variables x and y with domains S and T, respectively, along with an open sentence $P(x, y)$, we may have

$$\forall x \in S(\exists y \in T\ P(x, y)). \tag{3.2}$$

This is an example of **nested quantifiers**, when a quantified statement is in the scope of another quantifier. The more usual way that (3.2) is written is

$$\forall x \in S\ \exists y \in T\ P(x, y), \tag{3.3}$$

which may be read as "For each $x \in S$ there exists $y \in T$ such that $P(x, y)$." We may substitute "all" or "every" for "each" without changing the meaning of the statement. We find the negation of (3.3) using Theorem 3.6 twice:

$$\begin{aligned}
\neg(\forall x \in S\ \exists y \in T\ P(x, y)) &\equiv \neg(\forall x \in S(\exists y \in T\ P(x, y))) \\
&\equiv \exists x \in S(\neg(\exists y \in T\ P(x, y))) \\
&\equiv \exists x \in S(\forall y \in T(\neg P(x, y))) \\
&\equiv \exists x \in S\ \forall y \in T(\neg P(x, y))
\end{aligned} \tag{3.4}$$

Similarly we have

$$\neg(\exists x \in S\ \forall y \in T\ P(x,y)) \equiv \forall x \in S\ \exists y \in T(\neg P(x,y))). \tag{3.5}$$

Developing strategies for determining the truth value of a quantified statement, including statements such as (3.3) that possess nested quantifiers, will occupy all sections of this chapter after this one.

Example 3.8. State the negation of the statement "For every integer m there exists an integer n such that $|m - 2n + 1| \leq 2$."

Solution. Defining the open sentence $P(m,n) :\ |m - 2n + 1| \leq 2$, symbolically the statement is $\forall m \in \mathbb{Z}\ \exists n \in \mathbb{Z}\ P(m,n)$. With the equivalency (3.4) we find the negation of this statement to be, symbolically, $\exists m \in \mathbb{Z}\ \forall n \in \mathbb{Z}(\neg P(m,n))$. Since $\neg P(m,n)$ states that $|m - 2n + 1|$ is *not* less than or equal to 2, or in other words $\neg P(m,n)$ is $|m - 2n + 1| > 2$, we may write $\exists m \in \mathbb{Z}\ \forall n \in \mathbb{Z}(\neg P(m,n))$ as

$$\exists m \in \mathbb{Z}\ \forall n \in \mathbb{Z}(|m - 2n + 1| > 2),$$

or in words,

There exists an integer m such that, for all integers n, $|m - 2n + 1| > 2$.

Alternatively we may write

There is an integer m such that $|m - 2n + 1| > 2$ for every integer n.

This last version may be best since it requires minimal punctuation. ■

3.3 – Proof By Contraposition

We recall that the contrapositive of a conditional statement is logically equivalent to the conditional statement:

$$P \to Q \equiv \neg Q \to \neg P.$$

From this equivalency we readily find that

$$\forall x \in S(P(x) \to Q(x)) \equiv \forall x \in S(\neg Q(x) \to \neg P(x)),$$

and so to prove the universal conditional statement $\forall x \in S(P(x) \to Q(x))$ we may just as well prove $\forall x \in S(\neg Q(x) \to \neg P(x))$. This technique is called **proof by contraposition**. Indeed, it is often far easier to prove the contrapositive of a conditional statement than the statement itself.

Example 3.9. Let n be an integer. Prove that if $9n - 5$ is even, then n is odd.

Preliminaries. The statement to prove is: "For any integer n, if $9n - 5$ is even then n is odd." That is,

$$\forall n \in \mathbb{Z}((9n - 5 \text{ is even}) \to (n \text{ is odd})). \tag{3.6}$$

To prove (3.6), we prove the contrapositive statement

$$\forall n \in \mathbb{Z}((n \text{ is not odd}) \to (9n - 5 \text{ is not even})).$$

Of course, an integer that is not odd or even must be even or odd, respectively, so we may rephrase the contrapositive statement as follows:

$$\forall n \in \mathbb{Z}((n \text{ is even}) \to (9n - 5 \text{ is odd})). \tag{3.7}$$

Using the fact that pq, $p + q$, and $p - q$ are integers whenever p and q are integers, we now prove statement (3.7). ■

Proof. Let $n \in \mathbb{Z}$ be arbitrary, and suppose that n is even. Then $n = 2k$ for some $k \in \mathbb{Z}$, so that

$$9n - 5 = 9(2k) - 5 = 18k - 5 = (18k - 6) + 1 = 2(9k - 3) + 1.$$

Thus $9n - 5 = 2\ell + 1$ for integer $\ell = 9k - 3$, which shows that $9n - 5$ is odd. This proves (3.7), and therefore the logically equivalent statement (3.6) is proven. ■

Example 3.10. Prove that if the product of two positive real numbers is greater than 100, then at least one of the numbers is greater than 10.

Preliminaries. To say at least one of two numbers a and b is greater than 10 is to say that either $a > 10$ or $b > 10$ must be the case. The statement to prove is: "For any positive real numbers a and b, if $ab > 100$, then either $a > 10$ or $b > 10$." Letting $\mathbb{R}^+$ denote the set of positive real numbers, the statement to prove is

$$\forall a \in \mathbb{R}^+ \forall b \in \mathbb{R}^+((ab > 100) \to ((a > 10) \vee (b > 10))). \tag{3.8}$$

To prove (3.8), we prove the contrapositive statement. Since $\neg(ab > 100) \equiv (ab \leq 100)$ and

$$\neg((a > 10) \vee (b > 10)) \equiv (a \leq 10) \wedge (b \leq 10)$$

the contrapositive is

$$\forall a \in \mathbb{R}^+ \, \forall b \in \mathbb{R}^+(((a \leq 10) \wedge (b \leq 10)) \rightarrow (ab \leq 100)). \tag{3.9}$$

To prove (3.9) we use the following fact from algebra: If real numbers u, v, x, and y are such that $0 < u \leq v$ and $0 < x \leq y$, then $ux \leq vy$.[1] ■

Proof. Let $a \in \mathbb{R}^+$ and $b \in \mathbb{R}^+$ be arbitrary, and suppose $a \leq 10$ and $b \leq 10$. Since $a > 0$ and $b > 0$, it follows that $ab \leq (10)(10)$, and hence $ab \leq 100$. This proves (3.9), and therefore the equivalent statement (3.8) is proven. ■

[1] Note that $0 < u \leq v$ and $0 < x \leq y$ imply $1 \leq v/u$ and $1 \leq y/x$, so that $1 \leq (v/u)(y/x)$, and therefore $ux \leq vy$ as claimed. Now all that the reader must believe is that whenever $s \geq 1$ and $t \geq 1$, then $st \geq 1$!

3.4 – Proof By Cases

For the next example recall that, given two sets A and B, to prove that $A = B$ is usually accomplished by showing both $A \subseteq B$ and $B \subseteq A$. It is common to call, say, the expression $A \subseteq B$ a **containment**, in which case $B \subseteq A$ is referred to as the **reverse containment**.

Example 3.11. Let A and B be sets. Prove that $(A - B) \cup (B - A) = (A \cup B) - (A \cap B)$.

Preliminaries. We will state the result to be proven as follows: "For any sets A and B, $(A - B) \cup (B - A) = (A \cup B) - (A \cap B)$." The proof will require showing both the containment

$$(A - B) \cup (B - A) \subseteq (A \cup B) - (A \cap B) \tag{3.10}$$

and the reverse containment

$$(A \cup B) - (A \cap B) \subseteq (A - B) \cup (B - A). \tag{3.11}$$

To prove (3.10) we demonstrate that if $x \in (A - B) \cup (B - A)$, then it necessarily follows that $x \in (A \cup B) - (A \cap B)$. To prove (3.11) we demonstrate that if $x \in (A \cup B) - (A \cap B)$, then it necessarily follows that $x \in (A - B) \cup (B - A)$. ■

Proof. We show (3.10) to start. Let $x \in (A - B) \cup (B - A)$. Then either $x \in A - B$ or $x \in B - A$. We consider these two cases separately.

Case 1. Suppose $x \in A - B$. Then $x \in A$ and $x \notin B$. Because $x \in A$ we have $x \in A \cup B$, and because $x \notin B$ we have $x \notin A \cap B$. Thus $x \in (A \cup B) - (A \cap B)$.

Case 2. Suppose $x \in B - A$. Then $x \in B$ and $x \notin A$. Because $x \in B$ we have $x \in A \cup B$, and because $x \notin A$ we have $x \notin A \cap B$. Thus $x \in (A \cup B) - (A \cap B)$.

In both cases we conclude that $x \in (A \cup B) - (A \cap B)$, and so (3.10) is proven.

Now we show the reverse containment (3.11). Let $x \in (A \cup B) - (A \cap B)$. Then $x \in A \cup B$ and $x \notin A \cap B$, so that either $x \in A$ or $x \in B$, but not both. Again we consider two cases.

Case 1. Suppose $x \in A$. Then $x \notin B$ since $x \notin A \cap B$, so that $x \in A - B$ and hence $x \in (A - B) \cup (B - A)$.

Case 2. Suppose $x \in B$. Then $x \notin A$ since $x \notin A \cap B$, so that $x \in B - A$ and hence $x \in (A - B) \cup (B - A)$.

In both cases we conclude that $x \in (A - B) \cup (B - A)$, and so (3.11) is proven. Therefore $(A - B) \cup (B - A) = (A \cup B) - (A \cap B)$. ■

4

Relations and Functions

4.1 – Relations

Definition 4.1. *Let A and B be sets. A* ***binary relation from*** *A* ***to*** *B is a subset of $A \times B$. A* ***binary relation on*** *A is a subset of $A \times A$. To write $a\,R\,b$ or $a \sim b$ means $(a,b) \in R$. To write $a \not R\, b$ or $a \nsim b$ means $(a,b) \notin R$.*

Since binary relations are the only kind of relation we will consider, we will refer to any binary relation as simply a **relation**.

Since the empty set $\varnothing$ is a subset of $A \times B$ for any sets A and B, it is clear from the definition that $\varnothing$ is a relation from A to B (though not a very interesting one). In general $\varnothing$ is called the **empty relation**.

Definition 4.2. *Let R be a relation on A. If $x\,R\,x$ for all $x \in A$, then R is* ***reflexive****. If, for all $x, y \in A$, $y\,R\,x$ whenever $x\,R\,y$, then R is* ***symmetric****. If, for all $x, y, z \in A$, $x\,R\,z$ whenever $x\,R\,y$ and $y\,R\,z$, then R is* ***transitive****. If, for all $x, y \in A$, $x = y$ whenever $x\,R\,y$ and $y\,R\,x$, then R is* ***antisymmetric****.*

Thus a relation R on A is reflexive if $\forall x \in A(x \sim x)$, symmetric if

$$\forall x \in A\, \forall y \in A(x \sim y \to y \sim x),$$

transitive if

$$\forall x \in A\, \forall y \in A\, \forall z \in A((x \sim y \wedge y \sim z) \to x \sim z),$$

and antisymmetric if

$$\forall x \in A\, \forall y \in A((x \sim y \wedge y \sim x) \to x = y).$$

As we have noted in §3.1, because $\forall x \in S(P(x))$ is logically equivalent to $\forall x(x \in S \to P(x))$, we find the statement $\forall x \in S(P(x))$ to be (vacuously) true whenever $S = \varnothing$. As a consequence, any relation R on $\varnothing$ must be reflexive, symmetric, transitive, and antisymmetric (indeed R must itself be the empty relation $\varnothing$). What is more, the empty relation $\varnothing$ on any *nonempty* set A is symmetric, transitive, and antisymmetric, *but is not reflexive.*

4.2 – Equivalence Relations

Definition 4.3. *A relation R on a set A is an* ***equivalence relation*** *if R is reflexive, symmetric, and transitive.*

Theorem 4.4. *If R is an equivalence relation on A and $x, y \in A$, then $[x] = [y]$ if and only if $x\,R\,y$.*

Proof. Suppose $[x] = [y]$. Since R is reflexive we have $x\,R\,x$, so that $x \in [x]$, and hence $x \in [y]$. Because $[y] = \{a \in A : a\,R\,y\}$, we conclude that $x\,R\,y$. Therefore if $[x] = [y]$, then $x\,R\,y$.

For the converse, suppose that $x\,R\,y$. Let $a \in [x]$, so that $a\,R\,x$. Since R is transitive, from $a\,R\,x$ and $x\,R\,y$ we obtain $a\,R\,y$, which implies $a \in [y]$ and hence $[x] \subseteq [y]$.

Now let $a \in [y]$, so that $a\,R\,y$. Since R is symmetric, from $x\,R\,y$ we obtain $y\,R\,x$, and then $a\,R\,y$ and $y\,R\,x$ imply that $a\,R\,x$ since R is transitive. That is, $a \in [x]$, so that $[y] \subseteq [x]$. Therefore if $x\,R\,y$, then $[x] = [y]$. ■

Theorem 4.5. *If R is an equivalence relation on $A \neq \varnothing$, then $\mathcal{P} = \{[a] : a \in A\}$ is a partition of A.*

Proof. Suppose R is an equivalence relation on $A \neq \varnothing$. Since R is reflexive, we find that $x \in [x]$ for all $x \in A$, and so all elements of $\mathcal{P}$ are nonempty.

Suppose $[x], [y] \in \mathcal{P}$ with $[x] \neq [y]$. Then $x \not R\, y$ by Theorem 4.4. Let $z \in [x] \cap [y]$ be arbitrary. Then $z\,R\,x$ and $z\,R\,y$. Since R is symmetric it follows that $x\,R\,z$ and $z\,R\,y$, and hence $x\,R\,y$ by the transitive property. But this contradicats $x \not R\, y$. We conclude that $[x] \cap [y] = \varnothing$, and therefore the elements of $\mathcal{P}$ are mutually disjoint.

Let $x \in \bigcup \mathcal{P}$, where by definition $\bigcup \mathcal{P} = \bigcup_{a \in A} [a]$. Then there exists some $a \in A$ such that $x \in [a]$, and since $[a] \subseteq A$, it follows that $x \in A$ and hence $\bigcup \mathcal{P} \subseteq A$. Conversely, if $x \in A$, then the fact that $x\,R\,x$ implies $x \in [x]$, so that $x \in \bigcup \mathcal{P}$ and hence $A \subseteq \bigcup \mathcal{P}$. Therefore $A = \bigcup \mathcal{P}$, and $\mathcal{P}$ is a partition of A. ■

4.5 – Cardinalities of Sets

Theorem 4.6. *If A and B are denumerable sets and $A \cap B = \varnothing$, then $A \cup B$ is denumerable.*

Proof. Suppose A and B are disjoint denumerable sets. Then there exist bijections $f : \mathbb{N} \to A$ and $g : \mathbb{N} \to B$. Define $h : \mathbb{Z} \to A \cup B$ by

$$h(n) = \begin{cases} f(-n) & \text{if } n \leq -1 \\ g(n+1) & \text{if } n \geq 0 \end{cases}$$

Let $x \in A \cup B$. If $x \in A$, then $f(k) = x$ for some $k \geq 1$ since f is onto, and since $-k \leq -1$ we have $h(-k) = f(k) = x$. If $x \in B$, then $g(k) = x$ for some $k \geq 1$ since g is onto, and since $k - 1 \geq 0$ we have $h(k-1) = g(k) = x$. Hence h is onto.

Suppose $h(k) = h(m) = x$, so that either $x \in A$ or $x \in B$, but not both since A and B are disjoint. If $x \in A$, then by the definition of h we have $h(k) = f(-k)$ and $h(m) = f(-m)$, so that $f(-k) = f(-m)$, and thus $k = m$ since f is one-to-one. If $x \in B$, then by the definition of h we have $h(k) = g(k+1)$ and $h(m) = g(m+1)$, so that $g(k+1) = g(m+1)$, and thus $k = m$ since g is one-to-one. Hence h is one-to-one, and we conclude that h is a bijection.

Finally, since $\mathbb{Z}$ is denumerable there is a bijection $\varphi : \mathbb{N} \to \mathbb{Z}$, and then $h \circ \varphi : \mathbb{N} \to A \cup B$ is a bijection since the composition of two bijections is a bijection. Therefore $|A \cup B| = |\mathbb{N}|$, and $A \cup B$ is denumerable. ■

7

Enumerative Combinatorics

7.3 – The Pigeonhole Principle

Recall ye olde ceiling function $\lceil\cdot\rceil : \mathbb{R} \to \mathbb{Z}$, defined by

$$\lceil x \rceil = \min\{n \geq x : n \in \mathbb{Z}\}$$

for each $x \in \mathbb{R}$, with the property that $x \leq \lceil x \rceil < x + 1$.

Theorem 7.1 (**Pigeonhole Principle**). *If n objects are placed in k boxes, then at least one box contains at least $\lceil n/k \rceil$ objects.*

Proof. Suppose n objects are placed in k boxes. Assume it is not the case that at least one box contains at least $\lceil n/k \rceil$ objects. Letting B_i be the set of objects in the ith box, then $|B_i| \leq \lceil n/k \rceil - 1$ for each $1 \leq i \leq k$. Now, using the property that $x \leq \lceil x \rceil < x + 1$ for any $x \in \mathbb{R}$, we have

$$\sum_{i=1}^{k} |B_k| \leq \sum_{i=1}^{k} \big(\lceil n/k \rceil - 1\big) < \sum_{i=1}^{k} \big((n/k + 1) - 1\big) = \sum_{i=1}^{k} (n/k) = n.$$

Thus the total number of objects placed in the k boxes is less than n, which is a contradiction. Therefore at least one box contains at least $\lceil n/k \rceil$ objects. ■

Theorem 7.2 (**General Pigeonhole Principle**). *Let $S \neq \varnothing$ be a finite set with partition $\{S_1, S_2, \ldots, S_k\}$, and suppose $n_i \in \mathbb{N}$ is such that $|S_i| \geq n_i$ for each $1 \leq i \leq k$. Then for any $A \subseteq S$ with*

$$|A| \geq 1 + \sum_{i=1}^{k} (n_i - 1)$$

there exists some $1 \leq i \leq k$ such that $|A \cap S_i| \geq n_i$.

Example 7.3. Each item in a fruit basket is an apple or banana or tangerine or pear or lime. What is the minimum number of pieces of fruit that must be in the basket to guarantee that there is at least one apple, or at least two bananas, or at least three tangerines, or at least four pears, or at least five limes?

Solution. Let S be the set of all pieces of fruit in a basket, so if the basket contains, say, three apples, seven bananas, five tangerines, nine pears, and four limes, then $|S| = 3+7+5+9+4 = 28$. In particular this means that we consider any two pieces of fruit, including two pieces of the same kind (such as two limes) as being different objects. Let S_1 be the set of apples in a basket, S_2 the set of bananas, S_3 the set of tangerines, S_4 the set of pears, and S_5 the set of limes. We emphasize that, at this stage, $\{S_i : 1 \leq i \leq 5\}$ is not necessarily a partition of S, since it is not required that a fruit basket contain all five kinds of fruit, and therefore it may be that $S_i = \varnothing$ for some i. Our problem is to determine the minimum positive value of $|S|$ in order to guaratee that one of the following is true: $|S_1| \geq 1$, or $|S_2| \geq 2$, or $|S_3| \geq 3$, or $|S_4| \geq 4$, or $|S_5| \geq 5$; that is, $|S_i| \geq i$ for at least one value $1 \leq i \leq 5$.

We employ the general pigeonhole principle starting with a fruit basket whose associated set S is such that $|S_i| \geq i$ for *every* $1 \leq i \leq 5$. For such a fruit basket we have $|S| \geq 1+2+3+4+5 = 15$, but as it turns out we do not need a basket with at least 15 pieces of fruit to ensure that $|S_i| \geq i$ is true for at least one value of i. Setting $n_i = i$ for each $1 \leq i \leq 5$, the general pigeonhole principle states that if $A \subseteq S$ is such that

$$|A| \geq 1 + \sum_{i=1}^{5}(i-1) = 1 + (0+1+2+3+4) = 11,$$

then $|A \cap S_i| \geq i$ for at least one $1 \leq i \leq 5$. This tells us that in any "subbasket" A consisting of 11 pieces of fruit chosen from the "basket" S of at least 15 pieces of fruit, there must be at least i pieces of fruit from the set S_i that are also in A. Hence the "subbasket" A must contain at least 1 apple, or at least 2 bananas, or at least 3 tangerines, or at least 4 pears, or at least 5 limes, precisely as desired. Therefore, in order to be assured of having at least 1 apple, or at least 2 bananas, or at least 3 tangerines, or at least 4 pears, or at least 5 limes, a fruit basket must contain a minimum of 11 pieces of fruit.

This answer can be understood in common sense terms by imagining drawing pieces of fruit from a vast cargo container filled with apples, bananas, tangerines, pears, and limes. What is the minimum number that must be drawn to be guaranteed of getting one apple, or two bananas, or three tangerines, or four pears, or five limes? We consider the worst-case scenario: drawing one piece of fruit after another and getting no apples, one banana, two tangerines, three pears, and four limes. This amounts to 10 pieces of fruit drawn, and so the 11th piece of fruit drawn cannot fail to satisfy one of our five criteria. ■

8
Discrete Probability

8.1 – An Introduction to Discrete Probability

Theorem 8.1. *For $n \geq 2$ let $S_1, \ldots, S_n$ be sample spaces with associated probability functions $p_1, \ldots, p_n$, respectively, and let $S = S_1 \times \cdots \times S_n$. For any $E = E_1 \times \cdots \times E_n \subseteq S$ define*

$$p(E) = \prod_{i=1}^{n} p_i(E_i) = p_1(E_1)p_2(E_2)\cdots p_n(E_n).$$

Then p is a probability function on S.

10
Graph Theory

10.1 – Types of Graphs

Undirected Graphs

By a **graph** G is meant a kind of discrete mathematical structure consisting of a set $V(G)$ of **vertices** and a set $E(G)$ of **edges**. Thus G is characterized as an ordered pair $(V(G), E(G))$, and we write $G = (V(G), E(G))$. If we write simply $G = (V, E)$, as we frequently will in situations when the vertex or edge set of G must be referred to a great many times in some discussion, then it's understood that V and E are the vertex set and edge set of G, respectively.

Each edge $e \in E(G)$ is associated with one or two vertices in $V(G)$, called the **endpoints** of e. If e has endpoints $u, v \in V(G)$, then e is said to **join** u and v. To say a vertex u is **joined to** (or **adjacent to**) a vertex v means there exists an edge having u and v as endpoints.[2] We say an edge e is **incident to** a vertex v if v is an endpoint of e. If the endpoints of e are both v, then e is called a **loop** at v. We adopt the view that every edge has two endpoints, with a loop being an edge whose endpoints are both associated with (one might say "located at") the same vertex.

An edge e is an **undirected edge** if its endpoints at vertices u and v are not ordered, which is to say there is no thought of e "starting" at one vertex and "ending" at the other. We say a graph G is an **undirected graph** if all of its edges are undirected. If $e \in E(G)$ has endpoints u and v, and there is no other edge in $E(G)$ having the same endpoints, then e may be identified with the (unordered) set $\{u, v\}$, and in accordance with custom any one of the symbols $\{u, v\}$, $\{v, u\}$, vu, or uv (instead of the less descriptive symbol e) may be used to denote the edge. If it so happens that $u = v$, so that e is a loop at v, then e may be denoted by $\{v\}$ or vv so long as there is not a second loop at v that must also be referred to in a discussion. It will be our practice to treat the terms "undirected graph" and "graph" as synonymous, a point we shall elaborate on later in the section.

Two edges in a graph G are **parallel** if they have the same endpoints, and any maximal set of parallel edges (which is a set containing *all* the edges that join two particular vertices) is called a **multi-edge**. If G has parallel edges joining vertices $u, v \in V(G)$, then the symbol

[2] We may also say that u is **connected to** v, or that u and v are **connected**, but we shall see later that these particular uses of the term "connected" are special cases of a more general idea.

$\{u, v\}$ or uv is taken to denote a multi-edge; that is, $\{u, v\}$ or uv denotes the set of all edges in G having u and v as endpoints, rather than any one particular edge with endpoints u and v. We generally wish to be able to distinguish any one edge in a graph from all others, which is the so-called "edges with own identity" approach. When every edge in a graph has its own identity—its own unique symbol—then we need not resort to multisets or fret about ambiguities arising. If vertices u and v have, say, k edges joining them, we might denote the edges by $e_1, e_2, \ldots, e_k$. These would be k distinct objects belonging to the set $E(G)$.

An **incidence function** for a graph G is a function ι that maps each edge $e \in E(G)$ to the set of vertices that e is incident to (i.e. its endpoint set). Thus if e has endpoints u and v, then $\iota(e) = \{u, v\} = uv$. More formally, then, a graph G may be characterized as an ordered triple $(V(G), E(G), \iota)$, with the incidence function ι being the informant that apprises us what endpoints each $e \in E(G)$ possesses. It will be our custom to present a graph G as an ordered triple (V, E, ι), as opposed to a pair (V, E), if there is need for an incidence function ι in a particular situation, or in order to make a precise definition.

Without vertices there can be no edges, in which case the **empty graph** results. A graph is **nonempty** if it has at least one vertex, and though a graph with vertices does not necessarily have to have edges, it would be a poor graph indeed that did not possess at least one. A **trivial graph** is a graph possessing one vertex and no edges. An **infinite graph** is a graph consisting of an infinite number of vertices or edges, and a **finite graph** has a finite number of vertices and edges. We shall only make a study of finite graphs.

It is natural to draw graphs by depicting vertices as points and edges as line segments or curves that join the points. Such a depiction is called a **drawing** of a graph. Every figure in this section features one or more drawings of graphs, sometimes with the vertices labeled. Edges may also be labeled, and an edge in a directed graph will have an arrow somewhere along its length indicating it direction.

Two graphs G_1 and G_2 are **equal** if $V(G_1) = V(G_2)$ and $E(G_1) = E(G_2)$, in which case we write $G_1 = G_2$. We give now definitions for three types of undirected graph, each type a subset of the next type.

Definition 10.1. *A **simple graph** is an undirected graph without loops or parallel edges.*

Thus if $G = (V, E)$ is a simple graph, then for any $u, v \in V$ with $u \neq v$ there exists at most one edge $e \in E$ with endpoints u and v. If such an edge e exists, then it may be denoted by

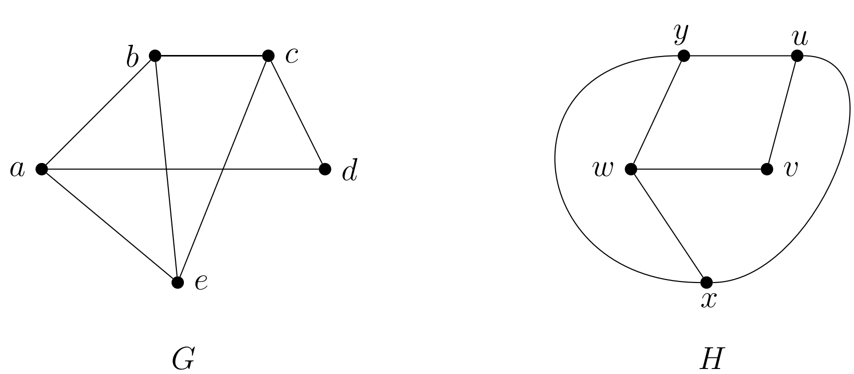


FIGURE 1. Drawings of two simple undirected graphs. Or are there really two?

the symbol $\{u, v\}$ or uv (instead of e) without ambiguity. The prohibition against loops means there exists no $e \in E$ such that $e = \{v\}$ for some $v \in V$. If ι is an incidence function for G, so that $G = (V, E, \iota)$, then for each edge $\{u, v\} \in E$ we have $\iota(\{u, v\}) = \{u, v\}$. Indeed, the incidence function of a simple graph is an identity function when edges are represented as sets of endpoints.

Example 10.2. In Figure 1 are drawings of two different simple graphs: $G = (V, E)$ at left and $H = (W, F)$ at right. But are these two graphs truly different mathemtically, in the way they join their respective vertices with edges? If the edges $\{y, x\}$ and $\{u, x\}$ in H were replaced with line segments, and v moved a bit to the right while w is moved a bit to the left, the drawing of H could be made identical to G save for the labeling of the vertices. Indeed, G has vertex set $V = \{a, b, c, d, e\}$ and edge set

$$E = \big\{\{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}, \{a, e\}, \{b, e\}, \{c, e\}\big\},$$

while H has vertex set $W = \{u, v, w, x, y\}$ and edge set

$$F = \big\{\{w, y\}, \{y, u\}, \{u, v\}, \{v, x\}, \{w, x\}, \{y, x\}, \{u, x\}\big\};$$

but if we were to relabel the vertices u, v, w, x, y of H as c, d, a, e, b, respectively, after making the aforementioned edge and vertex adjustments to the drawing of H, we could make the drawing of H perfectly identical to the drawing of G. In §10.4 we shall make precise the idea that two graphs with very different-looking drawings may be the same, or "isomorphic." ■

Definition 10.3. *A **multigraph** is an undirected graph that may have parallel edges but no loops.*

A more formal definition presents a multigraph as an ordered triple $G = (V, E, \iota)$, with incidence function $\iota : E \to \{\{u, v\} : u, v \in V \text{ and } u \neq v\}$ having a codomain that excludes any possibility of loops. Edges $e_1, e_2 \in E$ are parallel edges having endpoints u and v if and only if $\iota(e_1) = \iota(e_2) = \{u, v\}$. The multi-edge associated with u and v is $\iota^{-1}(\{u, v\})$, the inverse image of $\{u, v\}$ under ι, and so in particular $e_1, e_2 \in \iota^{-1}(\{u, v\})$. All the edges in $\iota^{-1}(\{u, v\})$ are edges in E whose endpoints are located at vertices u and v, with $u \neq v$ since a multigraph can have no loops.

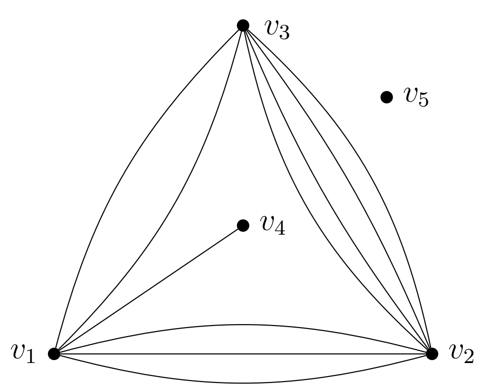


Figure 2. A multigraph. No edge is incident to vertex v_5.

Example 10.4. Figure 2 illustrates a multigraph $G = (V, E)$, with $V = \{v_1, v_2, v_3, v_4, v_5\}$ and E consisting of 10 edges. Letting e_{ij}^k denote the kth edge joining v_i to v_j, we could write

$$E = \{e_{12}^1, e_{12}^2, e_{12}^3, e_{13}^1, e_{13}^2, e_{23}^1, e_{23}^2, e_{23}^3, e_{23}^4, e_{14}^1\}.$$

With this sort of labeling scheme we could characterize the multigraph as $G = (V, E, \iota)$, where the incidence function ι is given by $\iota(e_{ij}^k) = \{v_i, v_j\}$ for all relevant values of i, j, k. For instance $\iota(e_{13}^1) = \iota(e_{13}^2) = \{v_1, v_3\}$. ■

If we wish to add loops to an undirected graph that is a simple graph or multigraph, then we obtain what is called a general graph.

Definition 10.5. *A* ***general graph*** *(or* ***pseudograph****) is an undirected graph that may have loops or parallel edges.*

As with a multigraph there is a more formal definition that characterizes a general graph as an ordered triple $G = (V, E, \iota)$, only now with incidence function $\iota : E \to \{\{u, v\} : u, v \in V\}$, so that the codomain of ι does not prohibit having $u = v$, and thus there may exist some $e \in E$ such that $\iota(e) = \{v\}$. Having $\iota(e) = \{v\}$ may be interpreted to mean that both endpoints of the edge e are located at the same vertex v (our preferred viewpoint), or that e has only one endpoint v (meh).

Graph theory being a relatively young field of mathematics, there is as yet no standard terminology, and so in particular many authors don't make a distinction between general graphs and multigraphs. Our practice will be to make little use of the terms multigraph and general graph, and instead refer to an undirected graph G as a "graph" if G is permitted to have loops or parallel edges, and a "simple graph" if G is forbidden to have loops or parallel edges.

Definitions , , and all contain the word "may." This is no accident: a multigraph *may* have parallel edges, but it does not have to; and a general graph *may* have parallel edges or loops, but it is not mandatory. As a result, the class of simple graphs is contained within the class of multigraphs, and the class of multigraphs is contained within the class of general graphs.

Types of Simple Graphs

The simple graph, which was defined first, is a type of graph that has many subtypes that are important in an introductory study of graph theory.

Definition 10.6. *A simple graph is a* ***complete graph on n vertices****, denoted by K_n, if $|V(K_n)| = n$ and for all $u, v \in V(K_n)$ with $u \neq v$ there exists exactly one $e \in E(K_n)$ such that $e = uv$. If a simple graph is not complete, then it is* ***noncomplete****.*

Thus every pair of distinct vertices in a complete graph are joined by exactly one edge. Illustrations of K_n for $1 \leq n \leq 6$ are supplied by Figure 3. The graph K_1 is trivially complete, for with there being but one vertex in the graph we find the statement "If $u \neq v$, then there exists exactly one edge $e \in E(K_1)$ such that $e = uv$" to be vacuously true for all $u, v \in V$.

How many edges does a complete graph on n vertices have? The following proposition gives the answer. Recall from our study of binomial coefficients that we define $C(n, k) = 0$ if $k > n$, and so in particular $C(1, 2) = 0$.

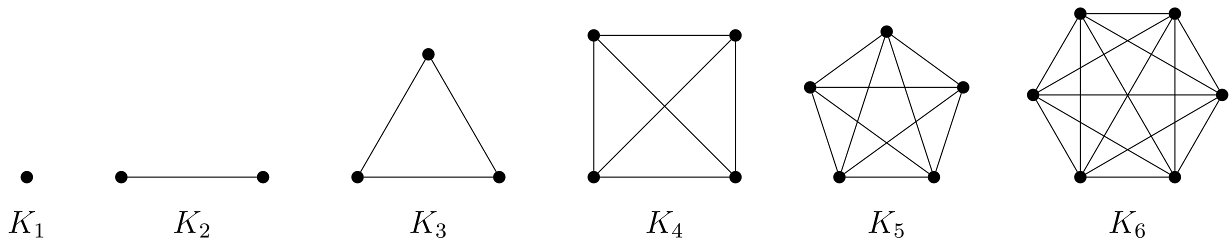


FIGURE 3. The complete graphs K_n for $1 \leq n \leq 6$.

Proposition 10.7. *For any $n \in \mathbb{Z}^+$, $|E(K_n)| = C(n, 2)$.*

Proof. The proof will be executed by induction. Clearly $|E(K_1)| = 0 = C(1, 2)$, which establishes the basis step. Suppose that $|E(K_n)| = C(n, 2)$ for some $n \geq 1$. The graph K_n has vertex set $V(K_n) = \{v_1, \ldots, v_n\}$, whereas $V(K_{n+1}) = \{v_1, \ldots, v_{n+1}\} = V(K_n) \cup \{v_{n+1}\}$. Thus K_{n+1} may be constructed from K_n by adding the vertex v_{n+1} to $V(K_n)$ to form $V(K_{n+1})$, and joining v_{n+1} to each of $v_1, \ldots, v_n$ with a single edge. This adds n edges to the edge set $E(K_n)$ to form $E(K_{n+1})$, and so $|E(K_{n+1})| = |E(K_n)| + n$. Employing our inductive hypothesis, we have

$$|E(K_{n+1})| = C(n, 2) + n = \frac{n!}{2!(n-2)!} + n = \frac{n(n+1)}{2} = \frac{(n+1)!}{2!(n-1)!} = C(n+1, 2),$$

and the proof is done. ∎

Definition 10.8. *A simple graph with $n \geq 3$ distinct vertices $v_1, v_2, \ldots, v_n$ is a **cycle**, denoted by C_n, if $E(C_n) = \{v_k v_{k+1} : 1 \leq k \leq n-1\} \cup \{v_n v_1\}$.*

For C_n as defined in Definition 10.8 it's not unheard-of to declare that $v_{n+1} = v_1$, so that E may be expressed more simply as $E = \{v_k v_{k+1} : 1 \leq k \leq n\}$. (To be sure, C_n still consists of precisely n vertices, but now one of them has dual designations.) The cycles C_3, C_4, C_5, and C_6 are shown in Figure 4

Definition 10.9. *Given a cycle C_n with vertex set $V' = \{v_1, \ldots, v_n\}$ and edge set*

$$E' = \{v_1v_2, v_2v_3, \ldots, v_{n-1}v_n, v_nv_1\},$$

*a simple graph (V, E) with $V = V' \cup \{v_0\}$ and $E = E' \cup \{v_0 v_k : 1 \leq k \leq n\}$ is a **wheel**, denoted by W_n.*

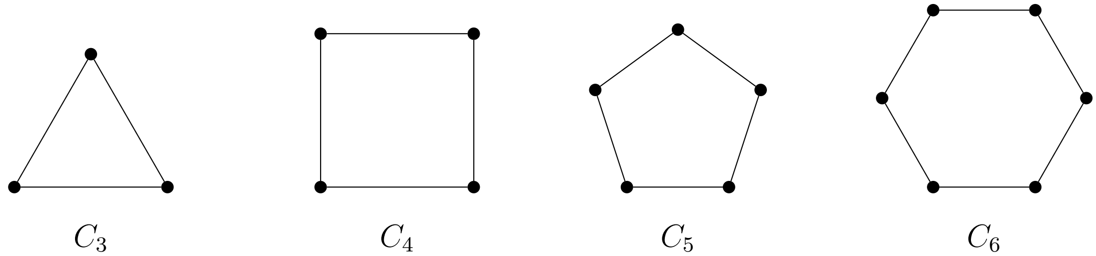


FIGURE 4. The cycles C_n for $3 \leq n \leq 6$.

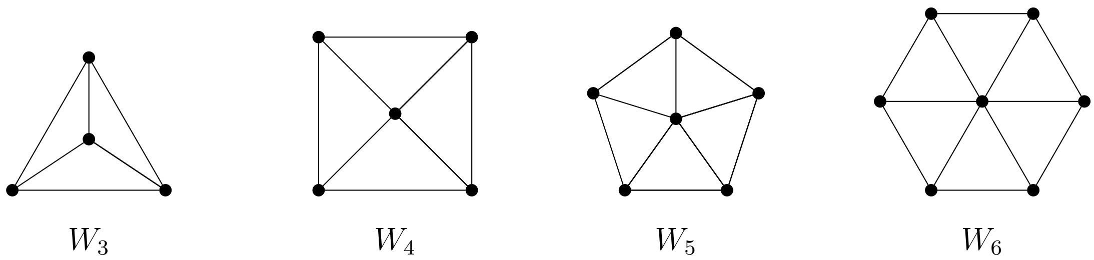


FIGURE 5. The wheels W_n for $3 \leq n \leq 6$.

In Figure 4 are shown the wheels W_3, W_4, W_5, and W_6. As Definition 10.9 indicates, a wheel W_n is created from the cycle C_n by adding one additional vertex and n edges. The vertex v_0 is most naturally placed at the center of a cycle, so that it becomes the "hub" of the resultant wheel, with the edges joining v_0 to all the n vertices of the cycle forming the "spokes" of the wheel.

One extremely useful kind of simple graph is the bipartite graph. Though we give a definition here, most of the theoretical results that we will develop in this chapter concerning bipartite graphs will appear in the next section and beyond.

Definition 10.10. *A simple graph G is **bipartite** (or a **bigraph**) if there exist $V_1, V_2 \subseteq V(G)$ such that $V_1 \cap V_2 = \varnothing$, $V_1 \cup V_2 = V(G)$, and every $e \in E(G)$ has one endpoint in V_1 and one endpoint in V_2. The pair (V_1, V_2) is a **bipartition** of $V(G)$.*

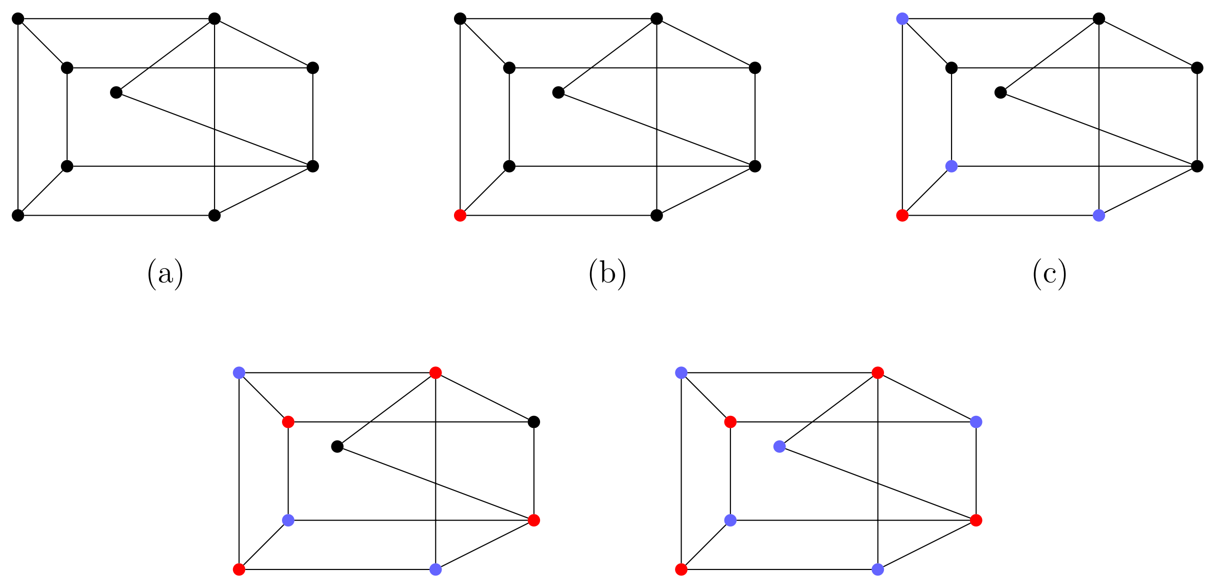
We observe that there is no prohibition against one or both sets in a bipartition (V_1, V_2) of $V(G)$ being the empty set, with $V_1 = V_2 = \varnothing$ implying that $V(G) = \varnothing$, in which case G is the empty graph. We further note that the definition of a bipartite graph may be written thus: "If the vertex set $V(G)$ of a simple graph G admits a bipartition, then G is bipartite." From this restatement of the definition it may be easier to perceive that the empty graph is trivially bipartite, as is the graph K_1 that has one vertex and no edges. Indeed a graph consisting of any number of vertices and no edges is bipartite, though such a graph has few uses.

Example 10.11. Any cycle with an even number of vertices is bipartite, which we now demonstrate. Let C_n be a cycle for which n is even, so that $n = 2m$ for an integer $m \geq 2$. The vertex set is $V = \{v_k : 1 \leq k \leq 2m\}$, and defining $v_{2m+1} = v_1$, the edge set is

$$E = \{v_k v_{k+1} : 1 \leq k \leq 2m\}.$$

Now define $V_1 = \{v_{2k} : 1 \leq k \leq m\}$ and $V_2 = \{v_{2k-1} : 1 \leq k \leq m\}$, so that V_1 is the set of even-indexed vertices and V_2 is the set of odd-indexed vertices. Clearly $V_1 \cap V_2 = \varnothing$ and $V_1 \cup V_2 = V$. Suppose $e \in E$, so that $e = v_k v_{k+1}$ for some $1 \leq k \leq 2m$. Clearly v_k and v_{k+1} must have opposite parity, even in the case when $k = 2m$, and so either $v_k \in V_1$ and $v_{k+1} \in V_2$ or the reverse. Therefore (V_1, V_2) is a bipartition of V and C_n is bipartite.

To show that C_n fails to be bipartite whenever $n \geq 3$ is odd is an exercise left to the reader. ∎



(a) (b) (c) (d) (e)

FIGURE 6. Using colors to algorithmically obtain a bipartition for a bipartite graph.

We give now a theorem that poses the question of whether a simple graph is bipartite in terms of whether the graph's vertices can be assigned one of two colors in such a way that no vertex is joined to another of the same color.

Theorem 10.12. *Let G be a simple graph. Then G is bipartite if and only if it is possible to assign one of two different colors to each vertex of G so that no edge joins two vertices of the same color.*

Proof. Suppose that $G = (V, E)$ is bipartite, and let (V_1, V_2) be a bipartition of V. By coloring all the vertices in V_1 red and all the vertices in V_2 blue, we find that no edge joins two vertices of the same color.

For the converse, suppose it's possible to color each vertex in V either red or blue in such a way that no edge joins two vertices of the same color. By letting V_1 be the set of all the red vertices and V_2 all the blue, we achieve a bipartition (V_1, V_2) of V and conclude that G is bipartite. ∎

Example 10.13. To determine whether the graph in Figure 6(a) is bipartite, we pick a vertex—any vertex—and assign it a color such as red, as in Figure 6(b). In Figure 6(c) all vertices joined to the red vertex are then colored blue, whereafter all uncolored vertices joined to the blue vertices are colored red in Figure 6(d). The process ends in Figure 6(e), when all vertices are either red or blue, and no two vertices of the same color are found to be adjacent. The graph is bipartite by Theorem 10.12, with the four red vertices and five blue vertices forming a bipartition of the vertex set of the graph.

If at any stage in such a procedure it is found to be impossible to assign a color to a particular vertex that does not avoid two like-colored vertices being joined, then the process stops and the graph in question is concluded to be not bipartite. ∎

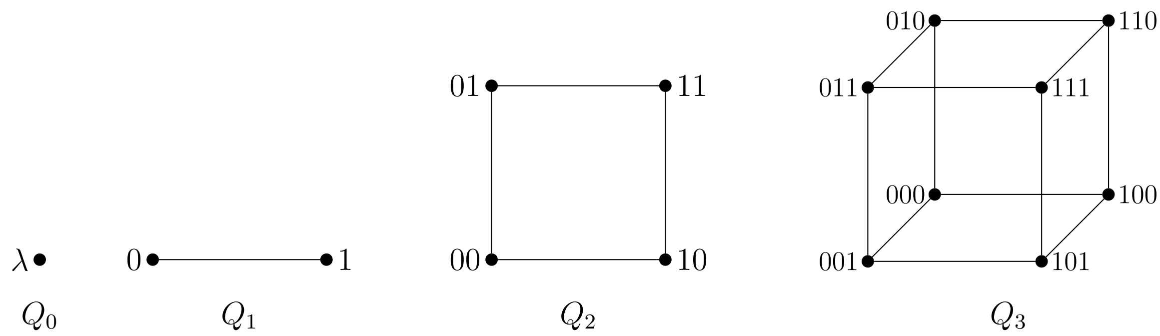


FIGURE 7.

We present one more special kind of simple graph in this section that is known as a hypercube graph, and to define it we recall the Hamming distance. Given two bitstrings of length n, $a = a_1a_2\cdots a_n$ and $b = b_1b_2\cdots b_n$, the Hamming distance between a and b is

$$\Delta(a,b) = \big|\{1 \le i \le n : a_i \ne b_i\}\big|;$$

that is, $\Delta(a,b)$ equals the number of values of the index i for which a_i and b_i do not match. We further recall that the empty string, denoted by λ, is the string of length 0.

Definition 10.14. *The **hypercube graph** Q_n is the simple graph for which $V(Q_n)$ is the set of all bitstrings of length n, and $E(Q_n) = \{uv : u,v \in V(Q_n)$ and $\Delta(u,v) = 1\}$. In the case when $n = 0$ we have $V(Q_0) = \{\lambda\}$ and $E(Q_0) = \varnothing$.*

The hypercube graphs Q_0 (a point), Q_1 (a line segment), Q_2 (a square), and Q_3 (a cube) are illustrated in Figure 7.

Proposition 10.15. *Any hypercube graph Q_n is bipartite.*

Proof. Let o denote the string of length n consisting only of zeros, and define

$$V_0 = \{v \in V(Q_n) : \Delta(o,v) \text{ is even}\} \quad \text{and} \quad V_1 = \{v \in V(Q_n) : \Delta(o,v) \text{ is odd}\}.$$

Clearly $V_0 \cup V_1 = V(Q_n)$ and $V_0 \cap V_1 = \varnothing$. To show that (V_0, V_1) is a bipartition of $V(Q_n)$, it remains to show that any $e \in E(Q_n)$ has one endpoint in V_0 and the other endpoint in V_1. This may be done by showing that no edge has both endpoints in V_0 or both endpoints in V_1.

Suppose u and v are distinct vertices in V_0, so that both $u = u_1\cdots u_n$ and $v = v_1\cdots v_n$ are bitstrings of length n that each have an even number of 1's (which includes the possibility of having no 1's and hence equaling o). Because $u \ne v$, there exists at least one index value $1 \le k \le n$ for which $u_k \ne v_k$, so that either $u_k = 1$ and $v_k = 0$, or $u_k = 0$ and $v_k = 1$. We may assume that $u_k = 1$ and $v_k = 0$ without loss of generality. Now, among the $n-1$ bits u_i for $i \ne k$ there must be an odd number of 1's, and among the $n-1$ bits v_i for $i \ne k$ there must be an even number of 1's. This implies that there must exist at least one index value $\ell \ne k$ for which $u_\ell \ne v_\ell$, and hence $\Delta(u,v) \ge 2$. In particular $\Delta(u,v) \ne 1$, so that $uv \notin E(Q_n)$, and therefore no two vertices in V_0 are adjacent. That this holds for V_1 as well is demonstrated by a similar argument. ∎

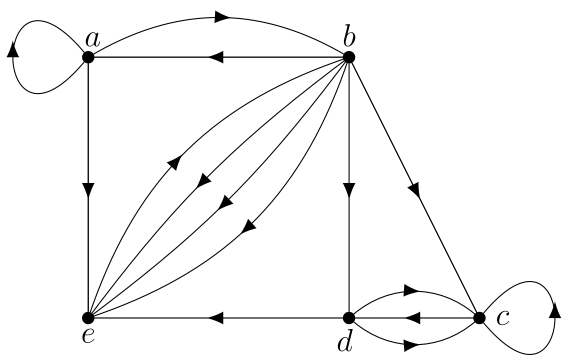


Figure 8.

Directed Graphs

A **directed graph**, or **digraph**, is a graph $D = (V, E)$ such that each edge has a particular **direction** (or **orientation**), and so is called a **directed edge** or **arc**. If arc $e \in E$ has endpoints $u, v \in V$, then e is associated with one of the ordered pairs (u, v) and (v, u), and we say that e **joins** u and v, or that u and v are **adjacent**. If e is associated with (u, v), then e is said to have **tail** u and **head** v, or that e **starts at** u and **ends at** v, and in a drawing of digraph D the arc e is denoted by an arrow pointing from u to v (hence the "tail" and "head" terminology). If D has only one arc with tail u and head v, then it is common practice to denote such an arc by (u, v) or uv. Two or more arcs in D having the same tail and the same head are called **parallel arcs**. For $u \neq v$, if arc e_1 has tail u and head v, and arc e_2 has tail v and head u, then e_1 and e_2 are *not* considered parallel, but rather are said to be **oppositely directed** (or **oppositely oriented**). If an arc's tail and head are both v, then the arc is a **loop at** v.

Generally our convention henceforth will be to use the word "graph" to mean "undirected graph." The term "graph theory" is an exception, as it encompasses the theory of both undirected and directed graphs. One other kind of graph that we will not make any study of is a **mixed graph**, which possesses both undirected and directed edges.

We give definitions for two types of digraph, included here so they may be readily contrasted with the different types of undirected graphs already defined. No theory or applications of digraphs will be entertained until §10.4.

Definition 10.16. *A **simple directed graph** (or **simple digraph**) is a digraph without loops or parallel arcs.*

As with the sundry types of undirected graphs, there is a more formal definition of a simple digraph D that incorporates an incidence function ι, and so characterizes D as an ordered triple $D = (V, E, \iota)$. The incidence function $\iota : E \to \{(u, v) : u, v \in V \text{ and } u \neq v\}$ is a one-to-one function such that $\iota(e) = (u, v)$ if $e \in E$ starts at u and ends at v. We observe that there may exist $e_1, e_2 \in E$ such that $\iota(e_1) = (u, v)$ and $\iota(e_2) = (v, u)$, but the codomain of ι compels $u \neq v$ since D cannot have loops, and certainly arcs e_1 and e_2 are not parallel since $\iota(e_1) \neq \iota(e_2)$. The endpoints of e_1 and e_2 are indeed the same two vertices (or "located" at the same two vertices), but e_1 has tail u and head v while e_2 has tail v and head u.

Definition 10.17. *A* ***directed multigraph*** *(or* ***multidigraph****) is a digraph that may have loops or parallel arcs.*

This definition of a multidigraph is worded so as to include all digraphs, including simple digraphs. Thus a multidigraph is not required to have loops or parallel arcs. A directed multigraph may be characterized as an ordered triple $D = (V, E, \iota)$, where $\iota : E \to V \times V$ is defined by $\iota(e) = (u, v)$ if $e \in E$ joins $u, v \in V$. An arc $e \in E$ is a loop at v if and only if $\iota(e) = (v, v)$. Arcs $e_1, \ldots, e_k \in E$ are parallel arcs if and only if $\iota(e_1) = \cdots = \iota(e_k)$.

10.2 – Undirected Graphs

In the previous section we defined what it means for two vertices in a graph to be joined, and it will be convenient to define a few related terms. Given an undirected graph G, we say vertices $u, v \in V(G)$ are **neighbors** (or **adjacent**) in G if u and v are endpoints of an edge $e \in E(G)$. Such an edge e is said to be **incident to** u and to v, and we may furthermore say that u is a **neighbor** of v (and v a neighbor of u). A vertex is its own neighbor if and only if it is joined to itself by a loop.

Definition 10.18. *Let G be an undirected graph. The set of all neighbors in G of a vertex $v \in V(G)$, denoted by $N_G(v)$, is called the **neighborhood** of v in G. For any $A \subseteq V(G)$, the **neighborhood** of A in G, denoted by $N_G(A)$, is the set*

$$N_G(A) = \bigcup_{v \in A} N_G(v).$$

If it does not give rise to ambiguity, the symbols $N(v)$ and $N(A)$ may be used instead of $N_G(v)$ and $N_G(A)$ to denote the neighborhood of $v \in V(G)$ and $A \subseteq V(G)$, respectively.

It can be seen that $v \in N(A)$ if and only if the vertex v is adjacent to at least one vertex in the set A.

Definition 10.19. *The **degree** of a vertex v in an undirected graph G, denoted by $d_G(v)$, is the number of edge endpoints located at v. We say v is **isolated** if $d_G(v) = 0$, and **pendant** if $d_G(v) = 1$.*

Again, if it does not occasion confusion, instead of $d_G(v)$ the symbol $d(v)$ may be used to denote the degree of a vertex v in a graph G.

In these notes we always regard an edge as having two endpoints, with a loop being an edge with both endpoints located at the same vertex. This makes the definition of the degree of a vertex v most elegant. Alas, many textbooks regard a loop as a one-ended edge, thereby turning terminology into foe instead of friend in the enterprise of expressing ideas clearly and succinctly. In such textbooks the degree of a vertex v must be defined as something along the lines of "the number of edges incident to v, except that a loop at v is counted twice." This alternative definition would also work for our purposes, but it is relatively cumbersome.

Example 10.20. We consider here the graph G in Figure 9, which is a general graph with vertex set $\{s, t, u, v, w, x, y, z\}$ and edge set consisting of 15 edges (including four loops). The degree of each vertex is: $d(s) = d(t) = d(u) = 6$, $d(v) = 5$, $d(w) = 3$, $d(x) = 0$, $d(y) = 4$, and $d(z) = 1$. Thus x is an isolated vertex and z is pendant. The neighborhood of each vertex is: $N(s) = \{s, t, w\}$, $N(t) = \{s, u, v, w\}$, $N(u) = \{t, u, v\}$, $N(v) = \{t, u, w\}$, $N(w) = \{s, t, v, z\}$, $N(x) = \varnothing$, $N(y) = \{y\}$, and $N(z) = \{w\}$. If $A = \{s, t, u\}$, then

$$N(A) = N(s) \cup N(t) \cup N(u) = \{s, t, u, v, w\}.$$

We note that a vertex is an element of its own neighborhood only when there is a loop at that vertex. ■

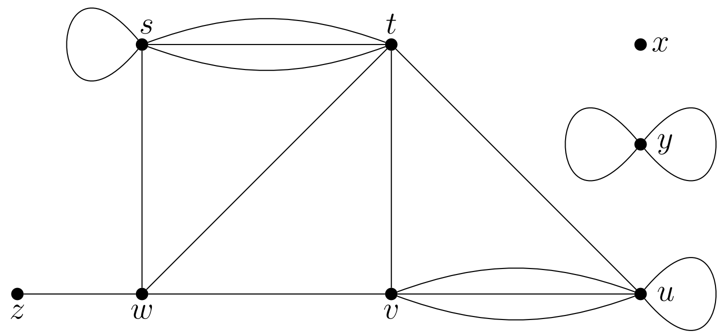


Figure 9.

We now state and prove a couple of theorems concerning the degrees of vertices in an undirected graph.

Theorem 10.21 (Handshaking Theorem). *If $G = (V, E)$ is an undirected graph, then*

$$2|E| = \sum_{v \in V} d(v). \tag{10.1}$$

Proof. Suppose $G = (V, E)$ is an undirected graph. Each $e \in E$ has two endpoints, and so must either contribute 1 to the degrees of two distinct vertices (if e is not a loop), or 2 to the degree of one vertex (if e is a loop). In either case e contributes 2 to the sum of the degrees of all vertices in V, so that the sum must equal twice the number of edges in E and the veracity of (10.1) is affirmed. ■

Theorem 10.22. *Any undirected graph has an even number of vertices of odd degree.*

Proof. Let $G = (V, E)$ be an undirected graph, V_o the set of vertices in G of odd degree, and V_e the set of vertices in G of even degree. Using Theorem 10.21, and observing that $V_o \cap V_e = \varnothing$ and $V_o \cup V_e = V$, we obtain

$$2|E| = \sum_{v \in V} d(v) = \sum_{v \in V_o} d(v) + \sum_{v \in V_e} d(v).$$

By definition any vertex $v \in V_e$ has even degree, so that $d(v) = 2n_v$ for some integer $n_v \geq 0$, and hence

$$\sum_{v \in V_o} d(v) = 2|E| - \sum_{v \in V_e} d(v) = 2\left(|E| - \sum_{v \in V_e} n_v\right).$$

This shows that $\sum_{v \in V_o} d(v)$ is an even integer, but because all the sum's terms are odd integers we conclude that the number of terms, which is $|V_o|$, must be even. Therefore G has an even number of vertices of odd degree. ■

There are a number of ways to create new graphs using one or more existing ones. First we need to be clear what it means for two graphs to be different. Given graphs G and H, we write $G = H$ if $V(G) = V(H)$ and $E(G) = E(H)$; otherwise we write $G \neq H$.

Definition 10.23. *A* ***subgraph*** *of $G = (V, E, \iota)$ is a graph $H = (W, F, \kappa)$ for which $W \subseteq V$, $F \subseteq E$, and $\kappa = \iota|_F$. We then write $H \subseteq G$ and say G* ***contains*** *H. If $H \subseteq G$ and $H \neq G$, then H is a* ***proper subgraph*** *of G and we write $H \subset G$.*

We see that H is a proper subgraph of G if and only if H is a subgraph of G and either $V(H) \subset V(G)$ or $E(H) \subset E(G)$. Any graph is a subgraph of itself, but not a proper one. Also any graph H that has more vertices or edges than a graph G cannot be a subgraph of G.

Example 10.24. Referring to Figures 4 and 5, it can be seen that the cycles C_n are proper subgraphs of the wheels W_n for $3 \leq n \leq 6$, and indeed this is the case for all $n \geq 3$. Comparing K_4 with W_4 in Figures 3 and 5, we see that K_4, despite having fewer edges and vertices than W_4, is not a subgraph of W_4 because K_4 has edges joining opposite corners of the square which W_4 lacks. Also W_4 may be recognized as not being a subgraph of K_4 for the simple reason that it has more vertices than K_4. ■

There are several types of subgraphs that are indispensable in the study of graph theory. The definitions we give here are what appear to be most common in the more modern literature, though many authors still steer by different stars, even among contemporary writers of mainstream discrete mathematics textbooks.

For the statement of the definitions we make use of a special notation to denote "travel" within a graph G along some subset of its edges. If a graph G has vertices $v_0, \ldots, v_n$, and for each $1 \leq i \leq n$ there is an edge e_i joining v_{i-1} to v_i, then it is possible to travel from v_0 to v_n along the edges $e_1, \ldots, e_n$, and we specify this route by writing the list $v_0 e_1 v_1 e_2 \cdots e_{n-1} v_{n-1} e_n v_n$, a finite alternating sequence of vertices and edges called a **vertex-edge sequence**. If G is a multigraph and e_1' is another edge with endpoints v_0 and v_1 (so that e_1 and e_1' are parallel edges), then the list $v_0 e_1' v_1 e_2 \cdots e_{n-1} v_{n-1} e_n v_n$ is considered to be a route from v_0 to v_n that is distinct from the former route that travels on e_1.

Definition 10.25. *Let $G = (V, E, \iota)$ be an undirected graph with incidence function ι, and let $n \in \mathbb{Z}^+$.*

A ***walk*** *is a subgraph of G consisting of a vertex-edge sequence $v_0 e_1 v_1 \cdots e_n v_n$ of vertices $v_0, \ldots, v_n \in V$ and edges $e_1, \ldots, e_n \in E$ such that $\iota(e_i) = \{v_{i-1}, v_i\}$ for all $1 \leq i \leq n$. A* ***trivial walk*** *is a walk consisting of a single vertex and no edges. A* ***subwalk*** *is any walk that is a proper subgraph of a walk.*

A ***trail*** *is a walk in which all edges are distinct. That is, if $v_0 e_1 v_1 \cdots e_n v_n$ is a trail, then $e_i \neq e_j$ whenever $i \neq j$.*

A ***path*** *is a trail in which all vertices are distinct. That is, if $v_0 e_1 v_1 \cdots e_n v_n$ is a path, then $v_i \neq v_j$ whenever $i \neq j$.*

The ***length*** *of a walk equals the number of edges (counting repetitions) in its associated sequence, so that $v_0 e_1 v_1 \cdots e_n v_n$ has length n. If $u = v_0$ and $v = v_n$, then $v_0 e_1 v_1 \cdots e_n v_n$ is a* ***u, v-walk****, with u being the* ***starting vertex*** *of the walk and v being the* ***ending vertex****. The* ***internal vertices*** *of a walk are those vertices that are neither the starting nor ending vertex.*

Definition 10.26. *A walk or trail is* ***closed*** *if its starting vertex and ending vertex are the same, otherwise it is* ***open****.*

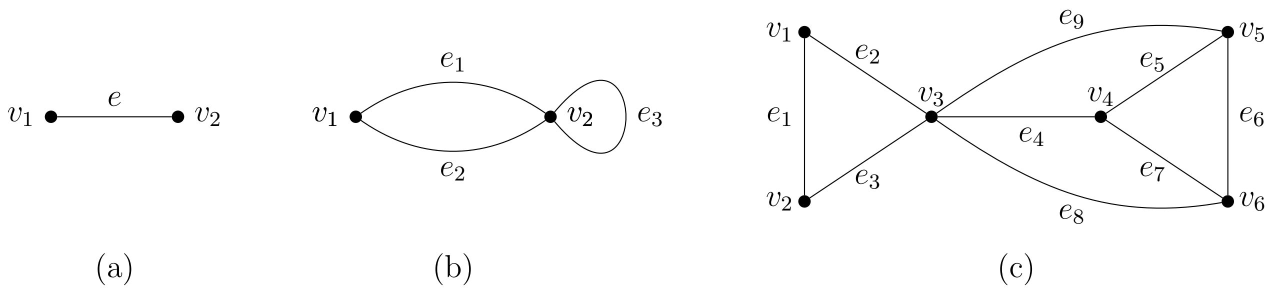


FIGURE 10.

A ***cycle*** *is a closed trail for which there exist no repeated internal vertices. A* k***-cycle*** *is a cycle of length* k*, an* ***odd cycle*** *is a cycle of odd length, and an* ***even cycle*** *is a cycle of even length. A* ***triangle*** *is a cycle of length* 3*.*

A ***circuit*** *is any closed trail. A* ***trivial circuit/cycle*** *is a circuit/cycle consisting of a single vertex and no edges.*

A u, v-walk is also known as a **walk from u to v**, and one may certainly speak also of a u, v-trail, u, v-path, or a trail or path from u to v. Also, if $u = v_0$ and $v = v_n$, then either of $v_0 e_1 v_1 \cdots e_n v_n$ or $v_n e_n \cdots v_1 e_1 v_0$ may be called a **walk between u and v**, with substitution of "trail" or "path" for "walk" permitted where appropriate. A walk is said to **pass through** any vertex in its vertex-edge sequence, and **traverse** any edge in its vertex-edge sequence.

A path may be defined as a *walk* in which all vertices are distinct, since having distinct vertices implies having distinct edges. There cannot be such a thing as a closed path, because to be closed requires the starting vertex of a walk $v_0 e_1 v_1 \cdots e_n v_n$, which is v_0, to equal the ending vertex v_n, and by definition a path cannot pass through any vertex more than once.

If a subgraph that is a cycle, as the term is defined by Definition 10.26, is viewed as being a graph in its own right (which it is), then it satisfies the properties of a cycle as defined by Definition 10.8. It can also be seen from Definitions 10.25 and 10.26 that a trivial walk, trivial cycle, and trivial circuit are all the same thing (a single vertex), and so we naturally regard a trivial walk to be a closed walk.

In a simple graph, which has no parallel edges, the **vertex sequence** $v_0 v_1 \ldots v_n$ associated with the vertex-edge sequence $v_0 e_1 v_1 \cdots e_n v_n$ of a walk is sufficient to fully define the walk. The same is true of the associated **edge sequence** $e_1 e_2 \ldots e_n$.

Example 10.27. We consider here the graph in Figure 10(a). The walk $v_1 e v_2$ is a path of length 1, and this path can just as well be specified by the vertex sequence $v_1 v_2$. The edge sequence e of the path, on the other hand, fails to specify which is the starting vertex, since the path $v_2 e v_1$ also has edge sequence e. The edge sequence of a walk is best used only in situations in which the vertex sequence is known.

The walk $v_1 e v_2 e v_1$ is neither a path (v_1 is repeated) nor a trail (e is repeated). Though the walk is closed and all internal vertices are distinct, it is not a cycle on account of it having a repeated edge. ■

Example 10.28. Consider the graph in Figure 10(b). The walk $v_2e_3v_2$ is a closed trail of length 1, and it is also a cycle since there are no repeated internal vertices. In this case the edge sequence e_3 is unambiguous with regards to which vertex is first, since v_2 is the only choice. The vertex sequence is v_2v_2. Technically v_2 is the vertex sequence of a walk of length 0, but this notion is excluded from Definition 10.25 since we shall have no need of it in the sequel.

The walk $v_1e_1v_2e_2v_1$ is a cycle of length 2, while $v_1e_1v_2e_3v_2e_2v_1$ is a closed trail—and hence a circuit—that is not a cycle since the internal vertices are not distinct. ■

A graph is **acyclic** if it contains no subgraph that is a cycle. Being defined as a kind of subgraph, a walk is itself a graph, and so we naturally define a walk to be **acyclic** if it has no subgraph that is a cycle.

Example 10.29. The graph in Figure 10(a) is acyclic, but the findings of Example 10.28 indicate that the graph in Figure 10(b) is not acyclic. The more complicated graph in Figure 10(c) is also not acyclic since, for instance, the subgraph that is the trail $v_3e_9v_5e_6v_6e_8v_3$ is a cycle. Since this cycle is a subgraph of the trail $v_3e_9v_5e_6v_6e_8v_3e_2v_1e_1v_2$, we conclude that the trail $v_3e_9v_5e_6v_6e_8v_3e_2v_1e_1v_2$ is not acyclic. The trail $v_5e_6v_6e_8v_3e_2v_1e_1v_2$ it acyclic, however. ■

Since a walk is a graph and our convention it to denote graphs by capital letters, we likewise may on occasion denote a walk by a capital letter such as W. Thus if W is a walk of length n, then $W = v_0e_1v_1 \cdots e_nv_n$.

Proposition 10.30. *Every closed walk $v_0e_1v_1 \cdots e_nv_n$ of length $n \geq 3$ with vertices $v_0, \ldots, v_{n-1}$ distinct is a cycle.*

Proof. Fix $n \geq 3$, and let $W = v_0e_1v_1 \cdots e_nv_n$ be a closed walk of length n with distinct vertices $v_0, \ldots, v_{n-1}$ in the graph $G = (V, E, \iota)$. Since $\iota(e_i) = \{v_{i-1}, v_i\}$ for $1 \leq i \leq n$, the edges $e_1, \ldots, e_{n-1}$ are distinct. We also have $\iota(e_n) = \{v_{n-1}, v_n\} = \{v_{n-1}, v_0\}$ since W is closed. Now, for $2 \leq i \leq n-1$ the edge e_i does not have v_0 as an endpoint, and so $e_i \neq e_n$. As for e_1, which has endpoints v_0 and v_1, the hypothesis $n \geq 3$ implies that $v_{n-1} \neq v_1$, and so $e_1 \neq e_n$. Thus the edges $e_1, \ldots, e_{n-1}$ are all distinct from e_n, so that W has distinct edges; and because W also has distinct internal vertices, we conclude that it is a cycle. ■

Proposition 10.31. *A walk is a path if and only if it is acyclic.*

Proof. We shall use induction to prove that, for any $n \in \mathbb{Z}^+$, if W_n is a path of length n then it is acyclic. The basis step is trivial: The path $W_1 = v_0e_1v_1$ is clearly acyclic. Fix $n \geq 1$, and suppose $W_{n+1} = v_0e_1v_1 \cdots e_{n+1}v_{n+1}$ is a path of length n. We observe that no cycle in W_{n+1} can contain v_{n+1}, since this vertex is joined only to v_n in W_{n+1}, and each vertex in a cycle must be adjacent to two other (distinct) vertices in the cycle. Thus any cycle in W_{n+1} must be a subgraph of $W_n = v_0e_1v_1 \cdots e_nv_n$. But W_n is a path of length n, and hence is acyclic by our inductive hypothesis. Therefore W_{n+1} is acyclic, and we have proven sufficiency (i.e. if a walk is a path then it is acyclic).

We now prove necessity by proving the contrapositive. Suppose a walk $W = v_0e_1v_1 \cdots e_nv_n$ is not a path, so that not all its vertices are distinct. Let $1 \leq j \leq n$ be the smallest integer for which the vertices in the list $v_0, \ldots, v_j$ are not distinct, so that there exists $0 \leq i < j$ such that

$v_i = v_j$ and $v_i, v_{i+1}, \ldots, v_{j-1}$ are distinct. Now, the walk $C = v_i e_{i+1} v_{i+1} \cdots e_j v_j$ is a subgraph of W for which all internal vertices are distinct, so that C is a cycle, and therefore W is not acyclic. ■

Theorem 10.32. *If there is a u, v-walk in a graph G with distinct vertices u and v, then there is a subwalk that is an acyclic u, v-path.*

Proof. For $u \neq v$, suppose there is a u, v-walk in a graph G. The length of the walk is at least 1, so that the set of lengths of all u, v-walks in G is a nonempty set of positive integers, and thus the well-ordering principle implies there is a u, v-walk $W = v_0 e_1 v_1 \cdots e_m v_m$ of minimum length $m \geq 1$, where $v_0 = u$ and $v_m = v$. Suppose there exist $1 \leq i < j \leq m$ such that $v_i = v_j$, so that W contains the closed subwalk $S = v_i e_{i+1} v_{i+1} \cdots e_{j-1} v_{j-1} e_j v_j$. (If $j = i + 1$ then S is a loop at v_i.) Deleting $e_{i+1} v_{i+1} \cdots e_{j-1} v_{j-1} e_j v_j$ from W yields the subwalk $W' = v_0 e_1 v_1 \cdots e_i v_i e_{j+1} v_{j+1} \cdots e_m v_m$. But W' is a u, v-walk that is shorter than W, which is impossible. Hence the vertices of W must be distinct, so that W is a u, v-path. By Proposition 10.31 this path is acyclic. ■

The next proposition could be proved in much the same manner as Theorem 10.32, but it should be more instructive to instead take a different approach.

Proposition 10.33. *If a closed trail passes through a vertex u in a graph G, then there is a subtrail that is a cycle passing through u.*

Proof. Let T be a closed trail in G with u as a vertex. If T is a loop at u, or if such a loop exists, then the loop is a cycle that passes through u and so there is nothing left to prove. We assume therefore that there is no loop at u. It will be convenient to cast u in the role of the first (and hence last) vertex of T, so that $T = u e_1 v_1 \cdots e_{n-1} v_{n-1} e_n u$. We must have $n \geq 2$, since if $n = 1$ we find $T = u e_1 u$ to be a loop at u. Finally, $v_{n-1} \neq u$ must be the case, otherwise T has subtrail $v_{n-1} e_n u = u e_n u$, again a loop at u.

Now, the subtrail $T' = u e_1 v_1 \cdots e_{n-1} v_{n-1}$ is a u, v_{n-1}-walk in G with $u \neq v_{n-1}$, and so Theorem 10.32 implies there is an acyclic u, v_{n-1}-path $u f_1 w_1 \cdots f_{m-1} w_{m-1} f_m v_{n-1}$ that is a subtrail of T'. Using this path we construct the circuit

$$C = u f_1 w_1 \cdots f_{m-1} w_{m-1} f_m v_{n-1} e_n u,$$

which is a subtrail of T. All the internal vertices of C are distinct, and so the subtrail C is a cycle passing through u. ■

Theorem 10.34. *If graph G has vertices $u \neq v$ and two different u, v-trails, then there is a cycle in G.*

Proof. Any loop in G is a cycle. Also, if there are parallel edges e_1 and e_2 joining any two distinct vertices u and v in G, then $u e_1 v e_2 u$ is a cycle in G. We henceforth assume that G is a simple graph, and so lacks loops and parallel edges.

Suppose G has vertices $u \neq v$ for which there exist two distinct u, v-trails T_1 and T_2. If either trail has a cycle then we're done, so assume T_1 and T_2 are acyclic, so that both are u, v-paths by Proposition 10.31. Since G is simple both paths are uniquely determined by their vertex sequence. Let $T_1 = u_0u_1\cdots u_n$ and $T_2 = v_0v_1\cdots v_m$, with $u_0 = v_0 = u$ and $u_n = v_m = v$.

That $T_1 \neq T_2$ implies there is some $i \geq 1$ such that $u_i \neq v_i$, so let $\ell = \min\{i : u_i \neq v_i\}$. That $u_n = v_m$ implies there exists $j \geq \ell$ and $k \geq \ell$ such that $u_j = v_k$, and we choose j and k such that $j + k$ is minimal. Let $w = u_{\ell-1} = v_{\ell-1}$ and $x = u_j = v_k$, and define the closed walk

$$C = u_{\ell-1}u_\ell\cdots u_jv_{k-1}v_{k-2}\cdots v_{\ell-1} = wu_\ell\cdots u_{j-2}u_{j-1}xv_{k-1}v_{k-2}\cdots v_\ell w.$$

The length of C is at least 3 since $u_\ell \neq v_\ell$, and we claim that C is a cycle. The u_i vertices in C are distinct since T_1 is a path, and the v_i vertices in C are distinct since T_2 is a path. Suppose for some $\ell - 1 \leq r \leq j$ and $\ell \leq s \leq k - 1$ we have $u_r = v_s$. But then $r + s \leq j + k - 1$, which contradicts the condition that j and k be chosen so $u_j = v_k$ and $j + k$ is minimal. Therefore all the vertices in C excluding $v_{\ell-1}$ are distinct, and by Proposition 10.30 we conclude that C is a cycle. ■

Definition 10.35. *Let $G = (V, E)$ be a graph with incidence function ι. The* ***subgraph induced by*** *$V' \subseteq V$ is the graph (V', E') with $E' = \{e \in E : \iota(e) \subseteq V'\}$.*

For $v \in V$, we denote by $G - v$ the subgraph induced by $V - \{v\}$, so that $G - v = (V - \{v\}, E')$ with E' consisting of those edges in E not incident to v. For $V' \subseteq V$, we denote by $G - V'$ the subgraph induced by $V - V'$.

More verbosely, given the graph $G = (V, E)$, the subgraph induced by $V' \subseteq V$ is the subgraph of G consisting of the vertices in V' and only those edges in E whose endpoints both lie in V'. Also from Definition 10.35 it can be seen that $G - v$ and $G - \{v\}$ are the same thing for any $v \in V$, with $G - v$ being merely a notational convenience. We have $G - v = (V - \{v\}, E)$ if v is an isolated vertex of G.

Example 10.36. Figure 5 depicts the wheels W_n for $3 \leq n \leq 6$. Let c be the center vertex of any wheel W_n. Referring to Figure 4, it can be seen that $W_n - c = C_n$. ■

Definition 10.37. *Let $G = (V, E)$ be a graph. If $e \in E$, then $G - e$ is the graph $(V, E - \{e\})$; and if $E' \subseteq E$, then $G - E'$ is the graph $(V, E - E')$. If $e \notin E$, then $G + e = (V, E \cup \{e\})$ provided the new edge e is designated to have endpoints that are in V.*

Ambiguities may arise with some of these notations. For instance, if u and v are vertices of a simple graph $G = (V, E)$ with $\{u, v\} \in E$, then $G - \{u, v\}$ could be the subgraph induced by $V - \{u, v\}$, or it could be the graph $(V, E - \{\{u, v\}\})$. The subgraph induced by $V - \{u, v\}$ eliminates from G not only the edge $\{u, v\}$, but also any other edge in E incident to u or v. Unless context makes clear which graph is meant by writing $G - \{u, v\}$, it may be better to denote an edge joining u and v by a symbol other than $\{u, v\}$, such as uv.[3]

Definition 10.38. *Let $\{G_i : 1 \leq i \leq n\}$ be a family of simple graphs, with edges in different graphs being considered identical if their endpoints are identically labeled. The* ***union*** *of $G_1, \ldots, G_n$ is the simple graph $\bigcup_{i=1}^n G_i$ defined by $\bigcup_{i=1}^n G_i = \big(\bigcup_{i=1}^n V(G_i), \bigcup_{i=1}^n E(G_i)\big)$.*

[3]Of course, uv could be the vertex sequence of a u, v-walk, so context is still important.

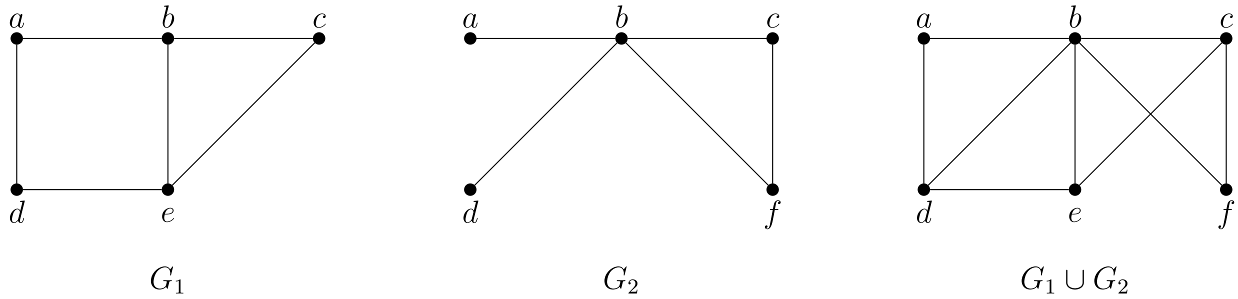


FIGURE 11.

Two graphs G and H are **disjoint** if $V(G) \cap V(H) = \varnothing$. If $\{G_i : 1 \leq i \leq n\}$ is a family of mutually disjoint graphs, so that $V(G_i) \cap V(G_j) = \varnothing$ whenever $i \neq j$, then the **disjoint union** of $G_1, \ldots, G_n$ is the graph $\bigsqcup_{i=1}^n G_i$ defined by $\bigsqcup_{i=1}^n G_i = \big(\bigsqcup_{i=1}^n V(G_i),\ \bigsqcup_{i=1}^n E(G_i)\big)$.

Example 10.39. We use here the notation uv to denote an edge $\{u, v\}$. If a graph G_1 has $V(G_1) = \{a, b, c, d, e\}$ and $E(G_1) = \{ab, ad, bc, be, ce, de\}$, and another graph G_2 has $V(G_2) = \{a, b, c, d, f\}$ and $E(G_2) = \{ab, bc, bd, bf, cf\}$, then the union $G_1 \cup G_2$ has

$$V(G_1 \cup G_2) = \{a, b, c, d, e, f\}$$

and

$$E(G_1 \cup G_2) = \{ab, ad, bc, bd, be, bf, ce, cf, de\}.$$

See Figure 11. We note that, for instance, the edge $ab \in E_1$ is considered identical to the edge $ab \in E_2$ since their endpoints are identically labeled as a and b. Thus we do not obtain parallel ab edges in the union of the graphs, and the union is a simple graph as a result. ■

Definition 10.40. *The **complement** of a simple graph G is the graph $\overline{G}$ with vertex set $V(\overline{G}) = V(G)$, and edge set $E(\overline{G})$ defined by $\{u, v\} \in E(\overline{G})$ if and only if $\{u, v\} \notin E(G)$.*

Definition 10.41. *The **distance** between vertices u and v, denoted by $d(u, v)$, is the length of the shortest possible u, v-path.*

The shortest possible path from v to v is the trivial path, which has length 0, and hence $d(v, v) = 0$.

Theorem 10.42. *If G is a nontrivial graph, then G is bipartite if and only if G has no odd cycles.*

Proof. Suppose nontrivial graph G is bipartite, with bipartition (V_1, V_2). Let $C = v_1 v_2 \cdots v_m v_1$ be a cycle in G, with $v_1 \in V_1$. Then $v_2 \in V_2$, $v_3 \in V_1$, $v_4 \in V_2$, and so on. In particular we have $v_i \in V_2$ if and only if i is even, and since v_m is adjacent to v_1, it follows that $v_m \in V_2$, and hence m is even. Since C has m edges, we conclude that C is an even cycle, and therefore G has no odd cycles.

Now suppose that nontrivial graph G has no odd cycles. We further assume, for now, that G is connected. For $u \in V(G)$, let V_1 consist of all $x \in V(G)$ such that $d(u, x)$ is even, and let

V_2 consist of all $y \in V(G)$ such that $d(u, y)$ is odd. Since $d(u, u) = 0$, we have $u \in V_1$. Clearly (V_1, V_2) is a partition of the set $V(G)$, but suppose that it is not a bipartition for G. Then either V_1 contains two adjacent vertices, or V_2 does.

We first consider the case wherein $v, w \in V_2$ are endpoints of an edge $e \in E(G)$. Then there exist integers $p, q \geq 0$ such that $d(u, v) = 2p+1$ and $d(u, w) = 2q+1$, so that there is a u, v-path $P_{uv} = v_0 v_1 \cdots v_{2p+1}$ (where $v_0 = u$ and $v_{2p+1} = v$) and a u, w-path $P_{uw} = w_0 w_1 \cdots w_{2q+1}$ (where $w_0 = u$ and $w_{2q+1} = w$). Let x be a vertex that lies on both paths, so that $x = v_i$. Now, because $d(u, v_i) < i$ would imply $d(u, v) < 2p + 1$, which is a contradiction, we must have $d(u, v_i) = i$. By the same token $d(u, w_i) = i$, and since w_i is the only vertex in P_{uw} that is a distance i from u, it follows that $x = v_i = w_i$. Certainly u is such a vertex x: $u = v_0 = w_0$. Let $m = \max\{i : v_i = w_i\}$, so that $v_m = w_m$ is the last common vertex in the vertex sequences for P_{uv} and P_{uw}. We now construct the cycle

$$C = v_m v_{m+1} \cdots v_{2p} v_{2p+1} w_{2q+1} w_{2q} \cdots w_{m+1} w_m.$$

The length of C is

$$[(2p + 1) - m] + 1 + [(2q + 1) - m] = 2(p + q - m + 1) + 1,$$

an odd number, and thus C is an odd cycle. As this is a contradiction, V_2 cannot contain adjacent vertices.

Now suppose $v, w \in V_1$ are adjacent. Then there exist $p, q \geq 0$ such that $d(u, v) = 2p$ and $d(u, w) = 2q$, so that there's a u, v-path $v_0 v_1 \cdots v_{2p}$ (where $v_0 = u$ and $v_{2p} = v$) and a u, w-path $w_0 w_1 \cdots w_{2q}$ (where $w_0 = u$ and $w_{2q} = w$). Again let v_m be the vertex farthest from u that the two paths share in common, so that $v_m = w_m$. We now construct the cycle

$$C = v_m v_{m+1} \cdots v_{2p} v_{2p} w_{2q} w_{2q} \cdots w_{m+1} w_m.$$

The length of C is $2(p + q - m) + 1$, again an odd cycle, again a contradiction. We conclude that (V_1, V_2) is indeed a bipartition, and therefore G is bipartite.

Finally, suppose nontrivial graph G is disconnected and has no odd cycles. Then each connected component of G has no odd cycles and thus must be bipartite by the preceding arguments, which implies that G itself is bipartite. ■

10.3 – Directed Graphs

Much of the basic terminology associated with directed graphs, or digraphs, was given in §10.1, but here we shall make mention of a few more terms. If D is a digraph with incidence function ι, distinct vertices $u, v \in V(D)$, and arc $e \in E(D)$ with endpoints u and v, then e has one of two possible directions (or orientations): e may **leave** u and **enter** v, so that $\iota(e) = (u, v)$; or e may leave v and enter u, so that $\iota(e) = (v, u)$. Being ordered pairs, we have $(u, v) \neq (v, u)$ whenever $u \neq v$. The tail and head of an arc are also called, respectively, the **initial vertex** and **terminal vertex** of the arc. An arc e is a loop at v if $\iota(e) = (v, v)$, and though in a drawing of a digraph a loop is drawn as a circular arrow, there is nothing in the machinery that we've developed to distinguish between a clockwise versus counterclockwise direction of "travel," and thus it is immaterial which way the arrow points.[4]

The following definition may be compared to Definition 10.25. As can be seen, the differences are slight.

Definition 10.43. *Let $D = (V, E, \iota)$ be a digraph with incidence function ι, and let $n \in \mathbb{Z}^+$.*

*A **directed walk** is a subgraph of D consisting of a **vertex-arc sequence** $v_0 e_1 v_1 \cdots e_n v_n$ of vertices $v_0, \ldots, v_n \in V$ and directed edges $e_1, \ldots, e_n \in E$ such that $\iota(e_i) = (v_{i-1}, v_i)$ for all $1 \leq i \leq n$. A **directed subwalk** is any directed walk that is a proper subgraph of a directed walk.*

*A **directed trail** is a directed walk in which all arcs are distinct, and a **directed path** is a directed trail in which all vertices are distinct.*

*The **length** of a directed walk equals the number of arcs in its associated sequence. If $u = v_0$ and $v = v_n$, then $v_0 e_1 v_1 \cdots e_n v_n$ is a **u, v-walk** of length n, with u being the **starting vertex** of the walk and v being the **ending vertex**. The **internal vertices** of a directed walk are those vertices that are neither the starting nor ending vertex.*

*A directed walk or trail is **closed** if its starting vertex and ending vertex are the same. A **directed cycle** is a closed directed trail for which there exist no repeated internal vertices. A **directed circuit** is any closed directed trail.*

As we've already done here, we will frequently refer to directed edges as arcs. Moreover, the word "directed" may be omitted whenever it will not occasion confusion, so that, for instance, a directed walk may simply be called a walk. Analogous to an undirected graph, in a digraph a **trivial walk/cycle/circuit** is defined as a walk/cycle/circuit with a single vertex and no arcs, with the absence of any arcs obviating the need to include the word "directed" in any of the three terms.

The definition of a directed walk $v_0 e_1 v_1 \cdots e_n v_n$ indicates that the direction of travel along such a walk must be consistent with the orientation of the walk's arcs: the head of arc e_i must coincide with the tail of arc e_{i+1}. Thus it is possible that a digraph has a u, v-walk but no v, u-walk.

In a simple digraph, which has no parallel arcs, the **vertex sequence** $v_0 v_1 \ldots v_n$ associated with the vertex-arc sequence $v_0 e_1 v_1 \cdots e_n v_n$ of a directed walk is sufficient to fully define the walk. The same is true of the associated **arc sequence** $e_1 e_2 \ldots e_n$.

[4]This is also why we do not speak of a "directed loop" here.

Definition 10.44. *Let D be a digraph with incidence function ι and $v \in V(D)$. The **in-neighborhood** or **predecessor set** of v is the set*

$$N_D^-(v) = \{x \in V(D) : \exists e \in E(D)\, [\iota(e) = (x, v)]\}.$$

*The **out-neighborhood** or **successor set** of v is*

$$N_D^+(v) = \{x \in V(D) : \exists e \in E(D)\, [\iota(e) = (v, x)]\}.$$

*Elements of $N_D^-(v)$ are the **in-neighbors** or **predecessors** of v in D, and elements of $N_D^+(v)$ are the **out-neighbors** or **successors** of v in D. We say v is a **source** if $N_D^-(v) = \varnothing$, and a **sink** if $N_D^+(v) = \varnothing$.*

Thus $N_D^-(v)$ is the set of all vertices in a digraph D that are the tail of an arc having head v, and $N_D^+(v)$ is the set of vertices in D that are the head of an arc having tail v.

Definition 10.45. *The **in-degree** of a vertex v in a digraph D, denoted by $d_D^-(v)$, is the number of arcs with head v; the **out-degree** of v, denoted by $d_D^+(v)$, is the number of arcs with tail v. We say v is a **source** if $d_D^-(v) = 0$, a **sink** if $d_D^+(v) = 0$ and **isolated** if $d_D^-(v) = d_D^+(v) = 0$.*

Example 10.46. Let D be the digraph with drawing given by Figure 12. The in-degree, out-degree, in-neighborhood, and out-neighborhood of the vertices in D are given in the following table.

v	$d_D^-(v)$	$d_D^+(v)$	$N_D^-(v)$	$N_D^+(v)$
a	2	3	$\{a, b\}$	$\{a, b, e\}$
b	2	6	$\{a, e\}$	$\{a, c, d, e\}$
c	4	2	$\{b, c, d\}$	$\{c, d\}$
d	2	3	$\{b, c\}$	$\{c, e\}$
e	5	1	$\{a, b, d\}$	$\{b\}$

Since $d_D^-(v) \neq 0$ and $d_D^+(v) \neq 0$ for all $v \in V(D) = \{a, b, c, d, e\}$, the graph has no sinks or sources. ■

If v is a source, then no arc has head at v; and if v is a sink, then no arc has tail at v. An isolated point is both a source and a sink.

Analogous to Theorem 10.21, which pertained to undirected graphs, we have the following for digraphs.

Theorem 10.47. *If $D = (V, E)$ is a digraph, then*

$$|E| = \sum_{v \in V} d_D^-(v) = \sum_{v \in V} d_D^+(v). \tag{10.2}$$

Proof. Each arc $e \in E$ has a tail at some vertex and a head at some vertex, and so contributes precisely 1 to the sum $\sum_{v \in V} d_D^-(v)$, and precisely 1 to the sum $\sum_{v \in V} d_D^+(v)$. Therefore each sum equals the total number of arcs. ■

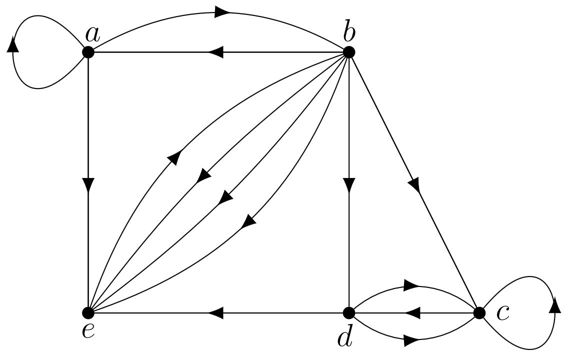


FIGURE 12.

As with undirected graphs, a digraph is **acyclic** if it contains no subgraph that is a cycle. Propositions 10.30 and 10.31, and also Theorem 10.32 and Proposition 10.33, apply equally well when all edges are directed rather than undirected. We repeat these results here—with wording included to emphasize their applicability to directed graphs—but omit the proofs for all but one since they are essentially the same as those given in §10.3. Indeed, in a u, v-walk W in an undirected graph, travel along W proceeds from u to v in a manner that forces each arc in the walk to be traversed in one particular direction, so that the arcs could just as well be directed arcs with orientations consistent with the direction of travel along W. What's different is only this: an undirected graph with a u, v-walk also has a v, u-walk, whereas in a digraph this symmetry may be broken.

Proposition 10.48. *In a digraph, every closed walk $v_0e_1v_1 \cdots e_nv_n$ of length $n \geq 3$ with vertices $v_0, \ldots, v_{n-1}$ distinct is a cycle.*

Proof. See the proof of Proposition 10.30. ■

Theorem 10.49. *If there is a u, v-walk in a digraph D with distinct vertices u and v, then there is a subwalk that is an acyclic u, v-path.*

Proof. See the proof of Theorem 10.32. ■

The next theorem is analogous to Proposition 10.33, but we furnish a proof here that is rather more concise.

Theorem 10.50. *If there is a closed walk from v to v in a digraph D, then there is a subwalk that is a cycle from v to v.*

Proof. If there is an arc $e \in E(D)$ that is a loop at v, then the closed walk vev is a cycle from v to v. Suppose D has no loop at v, and let $W = ve_1v_1 \cdots e_{n-1}v_{n-1}e_nv$ be a closed walk from v to v. We observe that $n \geq 2$ must be the case, for if $n = 1$ we would obtain $W = ve_1v$, which is a loop at v. Also we must have $v_{n-1} \neq v$, otherwise W has subwalk $v_{n-1}e_nv = ve_nv$, again a loop at v. Now, since D contains a v, v_{n-1}-walk for $v \neq v_{n-1}$, Theorem 10.49 implies that there is an acyclic v, v_{n-1}-path $P = vf_1u_1 \cdots f_{m-1}u_{m-1}f_mv_{n-1}$ in D, and then Pe_nv (the path P with arc e_n and vertex v tacked onto v_{n-1}) is a cycle from v to v. ■

Proposition 10.51. *In a digraph D, a walk is a path if and only if it is acyclic.*

Proof. See the proof of Proposition 10.31. ∎

We now turn our attention back to some concepts and theoretical developments without parallel in previous sections.

Definition 10.52. *Let D be a digraph with $u, v \in V(D)$. The* ***reverse*** *of an arc $e \in E(D)$ with tail u and head v is the arc denoted by e^{R} with tail v and head u. The* ***reverse*** *of a digraph D is the digraph D^{R} with $V(D^{\mathsf{R}}) = V(D)$ and $E(D^{\mathsf{R}}) = \{e^{\mathsf{R}} : e \in E(D)\}$.*

The reverse of a digraph D is also known as the **converse** or **transpose** of D. To obtain D^{R}, one simply reverses the direction of every arc in D. It is important to view the symbol R in Definition 10.52 as representing a mapping $e \mapsto e^{\mathsf{R}}$. In a simple digraph in which arcs may be uniquely identified with their endpoints, we find for any arc (u, v) that $(u, v)^{\mathsf{R}} = (v, u)$. The following proposition establishes a few facts concerning the reverse of a digraph.

Proposition 10.53. *Let D be a digraph.*
1. $(D^{\mathsf{R}})^{\mathsf{R}} = D$.
2. *D^{R} is acyclic if and only if D is acyclic.*
3. *For each $v \in V(D)$, $d^-_{D^{\mathsf{R}}}(v) = d^+_D(v)$ and $d^+_{D^{\mathsf{R}}}(v) = d^-_D(v)$*

Proof.
Proof of (1). If $e = (u, v)$, then $e^{\mathsf{R}} = (v, u)$, and thus $(e^{\mathsf{R}})^{\mathsf{R}} = (u, v) = e$. Now, $e \in E(D)$ iff $e^{\mathsf{R}} \in E(D^{\mathsf{R}})$ iff $e = (e^{\mathsf{R}})^{\mathsf{R}} \in E((D^{\mathsf{R}})^{\mathsf{R}})$, and so $E((D^{\mathsf{R}})^{\mathsf{R}}) = E(D)$. Since it's also the case that $V((D^{\mathsf{R}})^{\mathsf{R}}) = V(D^{\mathsf{R}}) = V(D)$, we conclude that $(D^{\mathsf{R}})^{\mathsf{R}} = D$.

Proof of (2). Suppose that D is not acyclic, so that it contains a cycle $v_0 e_1 v_1 \cdots e_{n-1} v_{n-1} e_n v_0$. Then D^{R} contains the cycle $v_0 e^{\mathsf{R}}_n v_{n-1} e^{\mathsf{R}}_{n-1} \cdots v_1 e^{\mathsf{R}}_1 v_0$, and so is itself not acyclic. Therefore if D^{R} is acyclic, then D is acyclic. This result in turn implies that if $(D^{\mathsf{R}})^{\mathsf{R}}$ is acyclic, then D^{R} is acyclic. Since $(D^{\mathsf{R}})^{\mathsf{R}} = D$ by part (1), the proof is done.

Proof of (3). Let $v \in V(D)$. Every arc with tail v in D is reversed so as to have head v in D^{R}, and every arc with head v in D is reversed so as to have tail v in D^{R}. Hence the number of arcs with head or tail v in D equals the number of arcs with tail or head v in D^{R}, respectively, and therefore $d^+_{D^{\mathsf{R}}}(v) = d^-_D(v)$ and $d^-_{D^{\mathsf{R}}}(v) = d^+_D(v)$. ∎

Proposition 10.54. *Every finite acyclic digraph has at least one sink and at least one source.*

Proof. We first show that there must be a sink. Let D be a finite acyclic digraph with $|V(D)| = n$, and pick a vertex $v_1 \in V(D)$. If v_1 is a sink, then we're done; and if v_1 is not a sink, then there is an arc e_1 from v_1 to another vertex $v_2 \neq v_1$. (If $v_2 = v_1$ then $v_1 e_1 v_1$ is a cycle, contradicting the acyclic property of D.) If v_2 is a sink, then we're done; and if not, then there is an arc e_2 from v_2 to some $v_3 \notin \{v_1, v_2\}$. This process continues: we're done if v_k is a sink, otherwise there is an arc e_k that leaves v_k and enters another vertex v_{k+1}. The walk $W = v_1 e_1 v_2 \cdots v_k e_k v_{k+1}$ is acyclic since it is a subgraph of an acyclic digraph, so that W is a path by Proposition 10.51, and thus the vertices $v_1, \ldots, v_{k+1}$ are distinct. Since there are only n vertices in $V(D)$, it is clear that this process must terminate at latest when we've constructed a path involving all n vertices, in which case the ending vertex v_n is a sink.

Now, D^{R} is also a finite acyclic digraph by Proposition 10.53(2), and so must have some vertex v that is a sink. This means that $d^+_{D^{\mathsf{R}}}(v) = 0$, and thus $d^-_D(v) = 0$ by Proposition 10.53(3). Therefore D has source v. ■

The edges in a digraph D being akin to one-way streets, it is possible that for $u, v \in V(D)$ there is a walk from u to v, but no walk from v to u.

Definition 10.55. *For a digraph D let $u, v \in V(D)$. Then v is **reachable from** u if there exists a directed u, v-walk in D, and u and v are **mutually reachable** if there is both a directed u, v-walk and a directed v, u-walk in D.*

10.4 – Graph Representations and Isomorphisms

Formally a graph G is an ordered triple (V, E, ι), with V being a set of vertices, E a set of edges, and ι a function that specifies the endpoints of each edge $e \in E$. This is an abstract mathematical structure, and as such, it can benefit greatly from alternative representations that are more visually informative, or more naturally encoded in a programming language. For visual appeal, nothing excels quite as well as a fully labeled drawing of a graph. To enter the information about a graph into a computer, however, matrices are most appropriate. In this section we shall consider two ways to represent an undirected or directed graph using matrices: an adjacency matrix and an incidence matrix.

Definition 10.56. *Let G be a graph with $V(G) = \{v_1, v_2, \ldots, v_n\}$, so that the vertices of G are ordered. The* ***adjacency matrix*** *of G is a matrix denoted by $\mathbf{A}_G$ with ij-entry $[\mathbf{A}_G]_{ij}$ defined as follows: if G is undirected, then $[\mathbf{A}_G]_{ij}$ equals the number of edges joining v_i and v_j; and if G is directed, then $[\mathbf{A}_G]_{ij}$ equals the number of edges having tail v_i and head v_j.*

Thus if G is an undirected graph with $|V(G)| = n$, then $\mathbf{A}_G$ will be a symmetric $n \times n$ matrix $\mathbf{A}_G$ with ij-entry equalling the number of edges joining the ith vertex in the sequence $v_1, v_2, \ldots, v_n$ to the jth vertex. The adjacency matrix $\mathbf{A}_G$ for a digraph G is not symmetric in general.

Example 10.57. In Figure 13 are the drawings of undirected graphs G, H, and K. For G we have $V(G) = \{v_1, v_2, v_3\}$, and so for $i, j \in \{1, 2, 3\}$ we shall let the ij-entry of $\mathbf{A}_G$ equal the number of edges joining v_i and v_j. Since $V(H) = \{v_1, v_2, v_3\}$, the same will be done to determine $\mathbf{A}_H$. Finally, for K we have $V(K) = \{a, b, c, d, e\}$, so it's only natural to order the vertices alphabetically, and thus $[\mathbf{A}_K]_{ij}$ will equal the number of edges joining the vertices labeled by the ith and jth letters of the alphabet. The adjacency matrices of the graphs are

$$\mathbf{A}_G = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \qquad \mathbf{A}_H = \begin{bmatrix} 0 & 2 & 0 \\ 2 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad \mathbf{A}_K = \begin{bmatrix} 1 & 2 & 0 & 0 & 1 \\ 2 & 0 & 1 & 1 & 4 \\ 0 & 1 & 1 & 3 & 0 \\ 0 & 1 & 3 & 0 & 1 \\ 1 & 4 & 0 & 1 & 0 \end{bmatrix}. \tag{10.3}$$

Only G is a simple graph, and the adjacency matrix of a simple graph is always a zero-one matrix.

The adjacency matrix of a graph is unique up to permutation of its rows and columns,[5] with each reordering of the rows and columns corresponding exactly to how the vertices in the graph's vertex set are reordered. For example, for the graph H we could let $V(H) = \{u_1, u_2, u_3\}$,

[5]The term "up to" is used in connection with declaring that the members of a collection of mathematical objects are the same or equal (or constitute a single unique object) in all respects except for one. Some examples: the symbols $\curlywedge$, $\succ$, $\curlyvee$, $\prec$ are identical up to rotation; the prime factorization of a number is unique up to the order of its factors ($6 = 2 \cdot 3$ versus $6 = 3 \cdot 2$); and the antiderivative of a function is unique up to an arbitrary constant. A final example: two n-dimensional vector spaces are the same up to isomorphism, so that essentially the spaces only differ with respect to the symbols used to denote their vectors.

where $(u_1, u_2, u_3) = (v_2, v_3, v_1)$, in which case we obtain

$$\mathbf{A}_H = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 0 & 0 \\ 2 & 0 & 0 \end{bmatrix},$$

which can be found either by relabeling the drawing of H in Figure 13 using u_1, u_2, u_3 in place of v_2, v_3, v_1, or rearranging the rows and columns of $\mathbf{A}_H$ in (10.3) in the following order: row 2 to row 1, row 3 to row 2, row 1 to row 3, column 2 to column 1, column 3 to column 2, and column 1 to column 3. We see that what's preserved in the two representations of $\mathbf{A}_H$ are the number of 0, 1, and 2 entries. ■

Because an adjacency matrix is a square matrix, it can be exponentiated by any positive integer, and it turns out that the powers of an adjacency matrix are themselves informative. For the proof of the next theorem we employ the notion of a concatenation of two walks in a graph, defined as follows: if W_1 is a v_0, v_ℓ-walk and W_2 is a $v_\ell, v_{\ell+m}$-walk, so that $W_1 = v_0 e_1 v_1 \cdots e_\ell v_\ell$ and $W_2 = v_\ell e_{\ell+1} v_{\ell+1} \cdots e_{\ell+m} v_{\ell+m}$, then the **concatenation** of W_1 and W_2 is the $v_0, v_{\ell+m}$-walk given by

$$W_1 * W_2 = v_0 e_1 v_1 \cdots e_\ell v_\ell e_{\ell+1} v_{\ell+1} \cdots e_{\ell+m} v_{\ell+m}.$$

Thus if one walk ends at the same vertex that another walk begins, then the two walks may be linked to form a longer walk.

Theorem 10.58. *Let G be a directed or undirected graph with $V(G) = \{v_1, \dots, v_m\}$. For every $n \in \mathbb{Z}^+$, $[\mathbf{A}_G^n]_{ij}$ equals the number of walks of length n from v_i to v_j for each $i, j \in \{1, \dots, m\}$.*

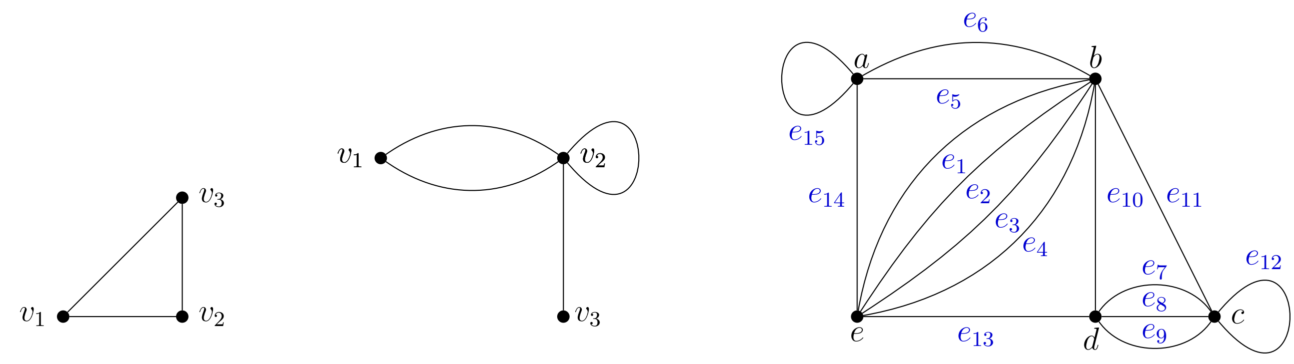
Proof. We adopt the view that either endpoint of an undirected edge may be designated the edge's tail, with the other endpoint then being the edge's head, so that the proof can address the cases when G is undirected or directed simultaneously. The proof will be by induction on the power n of $\mathbf{A}_G$.

Let $i, j \in \{1, \dots, m\}$. Since $[\mathbf{A}_G^1]_{ij} = [\mathbf{A}_G]_{ij}$, by definition $[\mathbf{A}_G^1]_{ij}$ equals the number of edges in $E(G)$ having tail v_i and head v_j, and so it equals the number of walks of length 1 from v_i to v_j. The basis step of the inductive argument is thus established.

Fix $n \in \mathbb{Z}^+$, and suppose $[\mathbf{A}_G^n]_{ij}$ equals the number of walks of length n from v_i to v_j for each $i, j \in \{1, \dots, m\}$. Fixing $i, j \in \{1, \dots, m\}$,

$$[\mathbf{A}_G^{n+1}]_{ij} = [\mathbf{A}_G^n \mathbf{A}_G]_{ij} = \sum_{k=1}^{m} [\mathbf{A}_G^n]_{ik} [\mathbf{A}_G]_{kj}. \tag{10.4}$$

by the definition of the matrix product. Now, for each $1 \le k \le m$, our inductive hypothesis implies that $[\mathbf{A}_G^n]_{ik}$ equals the number of walks of length n from v_i to v_k, and $[\mathbf{A}_G]_{kj}$ equals the number of walks of length 1 from v_k to v_j. If there exists a walk W_{ik} of length n from v_i to v_k, and also a walk W_{kj} of length 1 from v_k to v_j, then the concatenation $W_{ik} * W_{kj}$ is a walk of length $n + 1$ from v_i to v_j. By the multiplication rule of counting, the product $[\mathbf{A}_G^n]_{ik} [\mathbf{A}_G]_{kj}$ equals the total number of walks of length $n + 1$ from v_i to v_j for which the penultimate vertex is v_k. Since $1 \le k \le m$, by the sum rule of counting it follows that the sum at right in (10.4), and hence $[\mathbf{A}_G^{n+1}]_{ij}$ itself, equals the total number of walks of length $n + 1$ from v_i to v_j. ■



G $\quad$ H $\quad$ K

FIGURE 13.

In addition to an adjacency matrix, another way to represent a graph is with an incidence matrix. The incidence matrix of an undirected graph is always a zero-one matrix, which can be convenient; however, for most graphs (undirected or directed) the incidence matrix is much bigger than the adjacency matrix.

Definition 10.59. *Let G be a graph with $V(G) = \{v_1, \ldots, v_n\}$ and $E(G) = \{e_1, \ldots, e_m\}$, so that both the vertices and edges of G are ordered. The **incidence matrix** of G is a matrix $\mathbf{M}_G$ with ij-entry $[\mathbf{M}_G]_{ij}$ defined as follows: If G is undirected, then*

$$[\mathbf{M}_G]_{ij} = \begin{cases} 1, & \text{if } v_i \text{ is an endpoint of } e_j \\ 0, & \text{otherwise;} \end{cases} \tag{10.5}$$

and if G is a loopless digraph, then

$$[\mathbf{M}_G]_{ij} = \begin{cases} 1, & \text{if } v_i \text{ is the tail of } e_j \\ -1, & \text{if } v_i \text{ is the head of } e_j \\ 0, & \text{otherwise.} \end{cases} \tag{10.6}$$

Example 10.60. We find the incidence matrix for the graph K in Figure 13, the edges of which are labeled for the purpose. In determining $\mathbf{M}_K$ using (10.5), it may help to let $(a, b, c, d, e) = (v_1, v_2, v_3, v_4, v_5)$, which does not alter the order of the vertices in $V(K) = \{a, b, c, d, e\}$. Since K has 5 vertices and 15 edges, $\mathbf{M}_K$ will be a 5×15 matrix:

$$\mathbf{M}_K = \begin{bmatrix} 0&0&0&0&1&1&0&0&0&0&0&0&0&1&1 \\ 1&1&1&1&1&1&0&0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&1&1&1&0&1&1&0&0&0 \\ 0&0&0&0&0&0&1&1&1&1&0&0&1&0&0 \\ 1&1&1&1&0&0&0&0&0&0&0&0&1&1&0 \end{bmatrix}.$$

The fact that edges e_{12} and e_{15} are loops is reflected in the fact that the 12th and 15th columns of $\mathbf{M}_K$ each have just one entry equalling 1. All other columns have precisely two entries equalling 1, indicating edges with endpoints located at distinct vertices. ∎

Example 10.2 in §10.1 considered the graphs G and H pictured in Figure 1. It was noted that despite their drawings appearing quite different, the two graphs were in some sense the same. This notion we now make precise for simple undirected graphs, and later we shall expand the notion to include graphs that are not simple. We recall that each edge in a simple graph may be uniquely denoted by a set containing its endpoints.

Definition 10.61. *Two simple undirected graphs G and H are **isomorphic**, written $G \simeq H$, if there exists a bijection $\varphi : V(G) \to V(H)$ such that $\{u, v\} \in E(G)$ if and only if $\{\varphi(u), \varphi(v)\} \in E(H)$, in which case we call φ an **isomorphism**. Two graphs are **nonisomorphic** if they are not isomorphic.*

If G_1 and G_2 are isomorphic, we may also say G_1 is **isomorphic to** G_2. There are many kinds of isomorphisms in mathematics (for instance two vector spaces may be isomorphic), and so sometimes the function φ described in Definition 10.61 is called a **graph isomorphism**.

The notion of a graph isomorphism φ gives rise to a binary relation Φ known as the **isomorphism relation**. If $\mathcal{G}$ is the collection of all simple undirected graphs, then the isomorphism relation Φ is precisely that subset of $\mathcal{G} \times \mathcal{G}$ consisting of all possible pairs of isomorphic graphs. That is,

$$\Phi = \{(G_1, G_2) \subseteq \mathcal{G} \times \mathcal{G} : G_1 \simeq G_1\}.$$

It is straightforward to verify (as we will soon enough) that Φ is an equivalence relation: for any $G, H, K \in \mathcal{G}$, we have $G \simeq G$ (reflexivity), $G \simeq H$ if and only if $H \simeq G$ (symmetry), and $G \simeq K$ whenever $G \simeq H$ and $H \simeq K$ (transitivity). Thus the isomorphism relation gives rise to equivalence classes. We call an equivalence class of graphs associated with an isomorphism relation an **isomorphism class**.

Theorem 10.62. *The isomorphism relation Φ is an equivalence relation.*

Proof. Throughout the proof we shall assume that G, H, and K are simple graphs, and make use of the notation uv to denote an edge $\{u, v\}$.

Let $\varphi : V(G) \to V(G)$ be the identity function, so $\varphi(v) = v$ for all $v \in V(G)$. Clearly φ is a bijection, and the trivial observation that $v_1v_2 \in E(G)$ if and only if $v_1v_2 \in E(G)$ establishes that φ is an isomorphism, so that $G \simeq G$, and thus $(G, G) \in \Phi$. The relation Φ is therefore reflexive.

Next, suppose that $(G, H) \in \Phi$, so $G \simeq H$ and there exists an isomorphism $\varphi : V(G) \to V(H)$. Since φ is bijective it has an inverse φ^{-1} that is also bijective. Let $v_1v_2 \in E(H)$. There exist $u_1, u_2 \in V(G)$ for which $v_k = \varphi(u_k)$ for $k \in \{1, 2\}$, so that $\varphi(u_1)\varphi(u_2) \in E(H)$, and thus $\varphi^{-1}(v_1)\varphi^{-1}(v_2) = u_1u_2 \in E(G)$ since φ preserves edges. Conversely, if $\varphi^{-1}(v_1)\varphi^{-1}(v_2) \in E(G)$, then the edge-preserving property of φ implies that $v_1v_2 = \varphi(\varphi^{-1}(v_1))\varphi(\varphi^{-1}(v_2)) \in E(H)$. Having shown that $v_1v_2 \in E(H)$ if and only if $\varphi^{-1}(v_1)\varphi^{-1}(v_2) \in E(G)$, we find φ^{-1} to be edge-preserving, hence an isomorphism, and thus $H \simeq G$. We conclude that $(H, G) \in \Phi$, and therefore Φ is symmetric.

Finally, suppose $(G, H), (H, K) \in \Phi$, so $G \simeq H$ and $H \simeq K$, and there exist isomorphisms $\varphi : V(G) \to V(H)$ and $\psi : V(H) \to V(K)$. Since φ and ψ are bijective, so too is $\psi \circ \varphi$. Since φ and ψ perserve edges, $u_1u_2 \in E(G)$ implies $\varphi(u_1)\varphi(u_2) \in E(H)$, which in turn implies

$$(\psi \circ \varphi)(u_1)(\psi \circ \varphi)(u_2) = \psi(\varphi(u_1))\psi(\varphi(u_2)) \in E(K).$$

As shown in the previous paragraph, φ^{-1} is an isomorphism, and so too is ψ^{-1}. Suppose $\psi(\varphi(u_1))\psi(\varphi(u_2)) \in E(K)$. Since ψ^{-1} and φ^{-1} are edge-preserving, we have

$$\varphi(u_1)\varphi(u_2) = \psi^{-1}(\psi(\varphi(u_1)))\psi^{-1}(\psi(\varphi(u_2))) \in E(H),$$

and then $u_1 u_2 = \varphi^{-1}(\varphi(u_1))\varphi^{-1}(\varphi(u_2)) \in E(G)$. Thus $\psi \circ \varphi : V(G) \to V(K)$ is an isomorphism, implying that $G \simeq K$, and so $(G, K) \in \Phi$. Therefore Φ is transitive. ■

The second paragraph of the proof established the following corollary, which was used even in the third paragraph.

Corollary 10.63. *Given simple graphs G and H, if $\varphi : G \to H$ is an isomorphism, then $\varphi^{-1} : H \to G$ is also an isomorphism.*

A property of a graph G is **isomorphic invariant** or **preserved by isomorphism** (or simply **preserved**) if all the graphs belonging to the isomorphism class of G have the same property. For instance, since a graph isomorphism is a bijection, the number of vertices of a graph is preserved by isomorphism; that is, if $G \simeq H$ are isomorphic, then $|V(G)| = |V(H)|$. The number of edges of a graph is also isomorphic invariant, so that $|E(G)| = |E(H)|$. Moreover, given an isomorphism $\varphi : V(G) \to V(H)$, a property of the vertices of G is said to be preserved by isomorphism if v and $\varphi(v)$ have the same property for each $v \in V(G)$. For instance, it happens that $d_H(\varphi(v)) = d_G(v)$ for all $v \in V(G)$, and so the degree of a vertex is preserved. Finally, if any relationship amongst vertices $v_1, \ldots, v_n \in V(G)$ remains the same amongst vertices $\varphi(v_1), \ldots, \varphi(v_n) \in V(H)$, respectively, then that relationship is preserved by isomorphims. For instance, the property that $\{\varphi(u), \varphi(v)\} \in E(H)$ for any $\{u, v\} \in E(G)$ demonstrates that adjacency relationships amongst vertices in G are preserved. This property, which is part of the definition of a graph isomorphism, is what is meant when an isomorphism is said to **preserve edges** or be **edge-preserving**.

We remark now on a matter concerning drawings of graphs having no labels; that is, the vertices and edges in the drawing have no designation such as a letter, number, or other symbol. Suppose we were making a study of the family of all unit circles in a rectangular coordinate system. Each circle would be distinguished from all others by the coordinates (x, y) of its center, where x and y could be any real numbers. Other than its location, each circle has the same properties as all the others (e.g. the ratio of the circumference to the radius). The coordinates of each circle's center is merely a label, and does not speak to the circle's innate properties, or structure. Thus, we could draw a single unit circle in a coordinate-free plane, and think of such an "unlabeled" circle as being representative of an "equivalence class" of all the unit circles inhabiting the coordinate system. In analogous fashion, we shall interpret an unlabeled drawing of a graph as being representative of an isomorphism class of graphs. The drawing exhibits the adjacency relationship between vertices, or **structure**, that all members of the class share in common. The members of the class each have labeled vertices and edges, with the vertices in particular conceivably being given rectangular coordinates that not only label them, but also situate them somewhere so as to generate an infinitude of different-looking graphs that nonetheless all still have the same structure.

Example 10.64. We consider the simple graphs $G = (V, E)$ and $H = (W, F)$ given in Figure 1 and discussed in Example 10.2. Define the function $\varphi : V \to W$ as follows:

$$\varphi(a) = w, \quad \varphi(b) = y, \quad \varphi(c) = u, \quad \varphi(d) = v, \quad \varphi(e) = x.$$

That φ is one-to-one and onto, and hence a bijection, can be seen by inspection. Referring to the edge sets E and F given in Example 10.2, we also see that $\{a, b\} \in E$ and $\{\varphi(a), \varphi(b)\} = \{w, y\} \in F$, $\{b, c\} \in E$ and $\{\varphi(b), \varphi(c)\} = \{y, u\} \in F$, $\{c, d\} \in E$ and $\{\varphi(c), \varphi(d)\} = \{u, v\} \in F$, and so on. Thus φ preserves edges, and so is an isomorphism. Therefore G is isomorphic to H, and we write $G \simeq H$. ■

Theorem 10.65. *Suppose G and H are simple undirected graphs, $G \simeq H$, and $\varphi : V(G) \to V(H)$ is an isomorphism. Then the following hold:*
1. $|V(G)| = |V(H)|$.
2. $|E(G)| = |E(H)|$.
3. *For all $v \in V(G)$, $d_G(v) = d_H(\varphi(v))$.*

Proof. Statement (1) of the theorem is immediate from the fact that $V(G)$ and $V(H)$ are finite sets and φ is a bijection, while statement (2) is immediate from the fact that $E(G)$ and $E(H)$ are finite sets and φ preserves edges.

We now prove statement (3). Fix $v \in V(G)$. Then $\{\varphi(v), \varphi(x)\} \in E(H)$ for every $x \in V(G)$ such that $\{v, x\} \in E(G)$, and hence $d_G(v) \leq d_H(\varphi(v))$. Also, since $\varphi^{-1} : V(H) \to V(G)$ is an isomorphism by Corollary 10.63, for every $y \in V(H)$ such that $\{\varphi(v), y\} \in E(H)$ we find that $\{v, \varphi^{-1}(y)\} = \{\varphi^{-1}(\varphi(v)), \varphi^{-1}(y)\} \in E(G)$, and hence $d_H(\varphi(v)) \leq d_G(v)$. Therefore $d_G(v) = d_H(\varphi(v))$. ■

Example 10.66. We consider the three simple undirected graphs G, H, and K in Figure 14, and ask whether any two of them (or all three) are isomorphic. All the graphs have 5 vertices and 7 edges, and while this is necessary for graphs to be isomorphic, it is not sufficient.

Suppose $\varphi : V(G) \to V(H)$ is an isomorphism, and $u \in V(G)$ is such that $\varphi(u) = v_2$. By statement (3) of Theorem 10.65 it follows that $d_G(u) = d_H(\varphi(u)) = d_H(v_2) = 4$. However, there is no vertex in G having degree 4, so that we have arrived at a contradiction. Therefore G and H are not isomorphic. Since $d_K(w_4) = 4$, we also find G and K to not be isomorphic for the same reason.

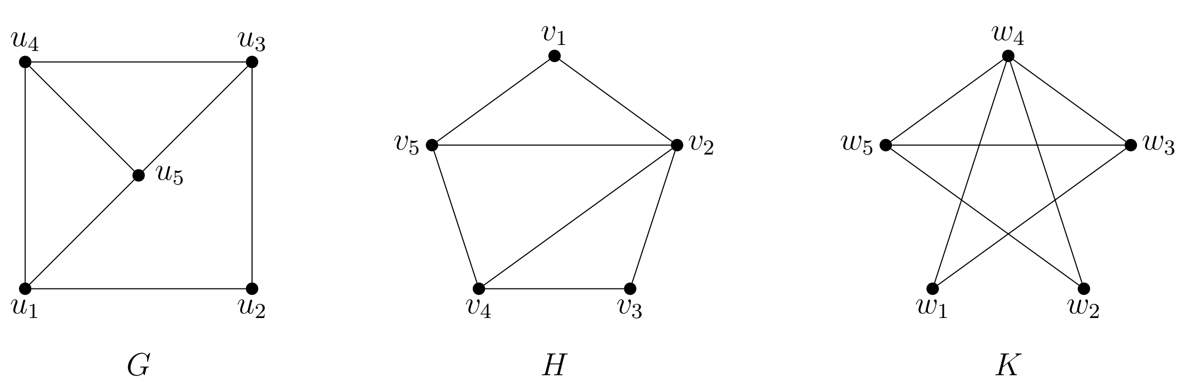


FIGURE 14.

Finally we come to H and K. Both of these graphs have one vertex of degree 4, two of degree 3, and two of degree 1. We begin to suspect that these graphs are isomorphic, and we attempt to construct an isomorphism $\varphi : V(H) \to V(K)$. Indeed, define $\varphi(v_1) = w_1$, $\varphi(v_2) = w_4$, $\varphi(v_3) = w_2$, $\varphi(v_4) = w_5$, $\varphi(v_5) = w_3$. The edge $v_1v_2 \in E(H)$ is preserved, since $\varphi(v_1)\varphi(v_2) = w_1w_4 \in E(K)$. The remaining six edges of H are likewise preserved, as can easily be checked. Since φ is found to be an edge-preserving bijection, and thus an isomorphism, we conclude that $H \simeq K$. ■

It can be shown that if the adjacency matrices $\mathbf{A}_G$ and $\mathbf{A}_H$ of simple graphs $G = (V, E)$ and $H = (W, F)$ are equal, then G and H are isomorphic. However, since the adjacency matrix of a graph with n vertices is uniquely determined only when the vertices have been ordered in some definitive way, such as $v_1, v_2, \ldots, v_n$, one ordering of the vertices of G or H may result in $\mathbf{A}_G = \mathbf{A}_H$, while another ordering may result in $\mathbf{A}_G \neq \mathbf{A}_H$. Thus G and H are not necessarily nonisomorphic if $\mathbf{A}_G \neq \mathbf{A}_H$.

Theorem 10.67. *Let $G = (V, E)$ and $H = (W, F)$ be simple graphs with $|V| = |W| = n$. If the vertices of G and H can be ordered so that $\mathbf{A}_G = \mathbf{A}_H$, then $G \simeq H$.*

Proof. Suppose the vertices of G and H can be ordered so that $\mathbf{A}_G = \mathbf{A}_H$. So, if we are initially given $V = \{v_1, \ldots, v_n\}$ and $W = \{w_1, \ldots, w_n\}$, we can find bijections $\sigma, \pi : [n] \to [n]$ such that, by letting $v'_i = v_{\sigma(i)}$ and $w'_i = w_{\pi(i)}$ for each $1 \leq i \leq n$, we obtain $\mathbf{A}_G = \mathbf{A}_H$ for $V = \{v'_1, \ldots, v'_n\}$ and $W = \{w'_1, \ldots, w'_n\}$.

Define $\varphi : V \to W$ by $\varphi(v'_i) = w'_i$, which is clearly a bijection.[6] For arbitrary $i, j \in [n]$, suppose $\{v'_i, v'_j\} \in E$. Then $[\mathbf{A}_H]_{ij} = [\mathbf{A}_G]_{ij} = 1$, so that $\{\varphi(v'_i), \varphi(v'_j)\} = \{w'_i, w'_j\} \in F$. Conversely, if $\{v'_i, v'_j\} \notin E$, then $[\mathbf{A}_H]_{ij} = [\mathbf{A}_G]_{ij} = 0$, and so $\{\varphi(v'_i), \varphi(v'_j)\} = \{w'_i, w'_j\} \notin F$. Hence φ preserves edges, is therefore an isomorphism, and we conclude that $G \simeq H$. ■

Reordering the vertices of graphs G and H in the manner done in the proof of Theorem 10.67 has the effect of rearranging both the rows and columns of $\mathbf{A}_G$ according to the permutation function σ, and of $\mathbf{A}_H$ according to the permutation function π. Thus if $\mathbf{A}_G$ and $\mathbf{A}_H$ are known, and it's possible to rearrange the rows and columns of, say, $\mathbf{A}_H$ according to some permutation function so as to obtain a matrix that is equal to $\mathbf{A}_G$, then it follows that $G \simeq H$.

To determine whether two simple graphs with an abundance of edges are isomorphic, it may be easier to consider the complements of the graphs and apply the following proposition.

Proposition 10.68. *Let G and H be simple graphs. Then $G \simeq H$ if and only if $\overline{G} \simeq \overline{H}$.*

Proof. Suppose $G \simeq H$, and let $\varphi : V(G) \to V(H)$ be an isomorphism. Then $\{u, v\} \in E(G)$ if and only if $\{\varphi(u), \varphi(v)\} \in E(H)$, so that $\{u, v\} \notin E(G)$ if and only if $\{\varphi(u), \varphi(v)\} \notin E(H)$, and hence $\{u, v\} \in E(\overline{G})$ if and only if $\{\varphi(u), \varphi(v)\} \in E(\overline{H})$. Since $V(\overline{G}) = V(G)$ and $V(\overline{H}) = V(H)$, we find $\varphi : V(\overline{G}) \to V(\overline{H})$ to be a bijection that preserves the edges, and hence is an isomorphism. Therefore $\overline{G} \simeq \overline{H}$.

[6] Being a bijection, σ has an inverse $\sigma^{-1} : [n] \to [n]$. Now, $\varphi(v'_i) = w'_i$ is equivalent to $\varphi(v_{\sigma(i)}) = w_{\pi(i)}$, and substituting $\sigma^{-1}(i)$ for i yields $\varphi(v_i) = w_{\pi(\sigma^{-1}(i))}$ for $1 \leq i \leq n$. This is an alternative formulation of φ that derives from not reordering the vertices of $V = \{v_1, \ldots, v_n\}$ at all (so $v'_i = v_i$) while letting $w'_i = w_{\pi(\sigma^{-1}(i))}$. It can be shown that this new scheme preserves the equality $\mathbf{A}_G = \mathbf{A}_H$.

The converse may be proven in much the same way, but we have already proven that $G \simeq H$ implies $\overline{G} \simeq \overline{H}$, and so $\overline{G} \simeq \overline{H}$ implies $\overline{\overline{G}} \simeq \overline{\overline{H}}$. Now merely note that $\overline{\overline{G}} = G$ and $\overline{\overline{H}} = H$. ■

The next proposition informs us that if one simple graph is bipartite and another isn't, then they cannot be isomorphic.

Proposition 10.69. *Suppose G and H are simple graphs and $G \simeq H$. Then G is bipartite if and only if H is bipartite.*

Proof. Suppose $G = (V, E)$ is bipartite, (V_1, V_2) is a bipartition of V, and $H = (W, F)$. Let $\varphi : V \to W$ be an isomorphism, and define $W_1 = \varphi(V_1)$ and $W_2 = \varphi(V_2)$. We show that (W_1, W_2) is a bipartition of W.

Suppose $w \in W_1 \cap W_2$. Then there exist $v_1 \in V_1$ and $v_2 \in V_2$ such that $\varphi(v_1) = \varphi(v_2) = w$, which violates the one-to-one property of φ, and therefore $W_1 \cap W_2 = \varnothing$.

Since W_1 and W_2 are subsets of W, we have $W_1 \cup W_2 \subseteq W$. Now suppose $w \in W$. Since φ is onto there exists some $v \in V$ such that $\varphi(v) = w$. Now, $v \in V_k$ for either $k = 1$ or $k = 2$, so that $w \in \varphi(V_k) = W_k$ and hence $w \in W_1 \cup W_2$. This demonstrates that $W \subseteq W_1 \cup W_2$, and therefore $W = W_1 \cup W_2$.

Next suppose that $\{w_1, w_2\} \in F$. Then there exist $v_1, v_2 \in V$ such that $\varphi(v_1) = w_1$ and $\varphi(v_2) = w_2$, and so $\{\varphi(v_1), \varphi(v_2)\} \in F$. Now, $\varphi^{-1} : W \to V$ preserves edges since it is itself an isomorphism, so that $\{v_1, v_2\} = \{\varphi^{-1}(\varphi(v_1)), \varphi^{-1}(\varphi(v_2))\} \in E$, and thus one endpoint of $\{v_1, v_2\}$ must be in V_1 and the other in V_2. For the sake of argument we assume that $v_k \in V_k$, from which $w_k \in \varphi(V_k) = W_k$ follows, and so one endpoint of $\{w_1, w_2\}$ is in W_1 and the other in w_2. Therefore (W_1, W_2) is a bipartition of W, and we conclude that H is bipartite.

The proof of the converse derives from the established conditional and the symmetry property of the isomorphism relation. ■

Definition 10.61 defines what it means for two simple undirected graphs to be isomorphic, and now we give a definition that applies to undirected graphs that are not necessarily simple (though they could be). For this more general definition of graph isomorphism we recall that an undirected graph G may be characterized as an ordered triple (V, E, ι), where $\iota : E \to \{\{u, v\} : u, v \in V\}$ is the incidence function of G.

Definition 10.70. *Let G and H be undirected graphs with incidence functions ι_G and ι_H, respectively. We say G and H are **isomorphic**, written $G \simeq H$, if there exist bijections $\varphi : V(G) \to V(H)$ and $\psi : E(G) \to E(H)$ such that, for all $u, v \in V(G)$ and $e \in E(G)$, $\iota_H(\psi(e)) = \{\varphi(u), \varphi(v)\}$ if and only if $\iota_G(e) = \{u, v\}$. Such a pair of functions (φ, ψ) is a **(graph) isomorphism**.*

In Definition 10.70 we have $\psi(e) \in E(H)$, and the endpoints of the edge $\psi(e)$ are the vertices $\varphi(u)$ and $\varphi(v)$ in $V(H)$ since $\iota_H(\psi(e)) = \{\varphi(u), \varphi(v)\}$. The property that $\iota_H(\psi(e)) = \{\varphi(u), \varphi(v)\}$ if and only if $\iota_G(e) = \{u, v\}$ is what is meant when it is said that a graph isomorphism (φ, ψ) **preserves edges** or **preserves incidence functions**.

As a notational convenience we shall frequently write $(\varphi, \psi) : G \to H$ to indicate that $\varphi : V(G) \to V(H)$ and $\psi : E(G) \to E(H)$, which after all could be written more compactly as $(\varphi, \psi) : (V(G), E(G)) \to (V(H), E(H))$, where $(V(G), E(G)) = G$ and $(V(H), E(H)) = H$.

It is a routine matter to verify that if $(\varphi, \psi) : G \to H$ is an isomorphism, then so too is $(\varphi^{-1}, \psi^{-1}) : H \to G$. If $(\varphi, \psi) : G \to H$ is an isomorphism and S is a subgraph of G, then the **image of S under** (φ, ψ) is defined to be the subgraph $(\varphi(V(S)), \psi(E(S)))$ of H.

10.5 – Graph Connectedness

Given an undirected graph G with distinct vertices u and v, the question may arise whether there exists a walk between u and v. More broadly, is it possible to walk from any vertex in G to any other vertex? Certainly the answer is no if G has at least two vertices and one is isolated, but even in the absence of isolated vertices the answer is not necessarily affirmative.

Definition 10.71. *Let G be an undirected graph. Two distinct vertices $u, v \in V(G)$ are* ***connected*** *if there exists a walk between them. The graph G is* ***connected*** *if u and v are connected for all $u, v \in V(G)$ such that $u \neq v$. A* ***connected component*** *of G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G. A connected component is* ***trivial*** *if it consists of one vertex and no edges (the trivial graph), otherwise it is* ***nontrivial****. If G is not connected, then it is said to be* ***disconnected****.*

From the definition we deduce that a graph G is disconnected if and only if there are vertices $u \neq v$ in $V(G)$ for which there exists no u, v-walk. From this it follows that any graph consisting of only one vertex is connected. The empty graph, which has no vertices, is connected in a vacuous sense. Any operation performed on a connected graph G that results in a new graph that is disconnected, such as deleting a vertex and its incident edges, is said to **disconnect** the graph G.

Proposition 10.72. *If G is a connected undirected graph, then for all $u \neq v$ in $V(G)$ there exists a u, v-path.*

Proof. Suppose G is connected and $u, v \in V(G)$ are distinct. Since G is connected it contains a u, v-walk, and therefore it contains a u, v-path by Theorem 10.32. ■

Typically a connected component of an undirected graph is called simply a component. As the next proposition establishes, no two components of a graph can have an edge or vertex in common, and thus any graph is a disjoint union of its components.

Proposition 10.73. *The connected components of an undirected graph G are mutually disjoint.*

Proof. Suppose that C_1 and C_2 are distinct components of graph G, and that $w \in V(C_1) \cap V(C_2)$. Fix $u, v \in V(C_1) \cup V(C_2)$, where $u \neq v$. If $u, v \in V(C_1)$ or $u, v \in V(C_2)$, then u and v are connected since C_1 and C_2 are connected. Suppose $u \in V(C_1)$ and $v \in V(C_2)$. Then there is a u, w-walk W_1 and a w, v-walk W_2, so that the concatenation $W_1 * W_2$ is a u, v-walk and thus u and v are again connected. It follows that $C_1 \cup C_2$ is a connected subgraph of G, and since C_1 cannot be a proper subgraph of another connected subgraph of G, we must have $C_1 = C_1 \cup C_2$ and hence $C_2 \subseteq C_1$. But C_2 itself cannot be a proper subgraph of another connected subgraph of G, so that we are forced to conclude that $C_1 = C_2$, which contradicts the hypothesis that C_1 and C_2 are distinct. Therefore $V(C_1) \cap V(C_2) = \varnothing$, so that C_1 and C_2 are disjoint. Since C_1 and C_2 are arbitrary components of G, the statement of the proposition follows. ■

The next proposition establishes that the property of being connected or disconnected is an isomorphic invariant.

Proposition 10.74. *If undirected graphs G and H are isomorphic, then they are either both connected or both disconnected.*

Proof. Suppose $G \simeq H$, and let (φ, ψ) be a graph isomorphism. Suppose G is connected, and let $v, \hat{v} \in V(H)$ with $v \neq \hat{v}$. Because $\varphi : V(G) \to V(H)$ is onto, there exist $u, \hat{u} \in V(G)$ such that $v = \varphi(u)$ and $\hat{v} = \varphi(\hat{u})$; and because φ is one-to-one, we have $u \neq \hat{u}$. Now, G being connected, there is a walk $u_0 e_1 u_1 \cdots u_{n-1} e_n u_n$ in G with $u_0 = u$ and $u_n = \hat{u}$. Let

$$W = \varphi(u_0)\psi(e_1)\varphi(u_1) \cdots \varphi(u_{n-1})\psi(e_n)\varphi(u_n),$$

and denote the incidence functions of G and H by ι_G and ι_H. Since $\iota_G(e_k) = \{u_{k-1}, u_k\}$ for each $1 \leq k \leq n$, we have $\iota_H(\psi(e_k)) = \{\varphi(u_{k-1}), \varphi(u_k)\}$ with $\varphi(u_0) = v$ and $\varphi(u_n) = \hat{v}$, and so W is a $v, \hat{v}$-walk in H. Therefore H is connected, and since a symmetrical argument will show that G is connected whenever H is, we conclude that G is connected if and only if H is connected. ■

Not only is the property of connectedness an isomorphic invariant, so too is the number of components that a graph possesses.

Proposition 10.75. *If undirected graphs G and H are isomorphic, then they have the same number of connected components.*

Proof. Suppose $G \simeq H$, and G has n connected components. If $n = 1$ then G is connected, whereupon Proposition 10.74 informs us that H is also connected, and so both G and H have one component. We henceforth assume that $n \geq 2$, so that G is disconnected, and hence so too is H.

Let $(\varphi, \psi) : G \to H$ be a graph isomorphism, let C be a component of G, and denote by D the image of C under (φ, ψ), so that $V(D) = \varphi(V(C))$ and $E(D) = \psi(E(C))$. Now, $C \simeq D$ since $(\varphi|_{V(C)}, \psi|_{E(C)}) : C \to D$ is an isomorphism, and so D is connected by Proposition 10.74. In particular this implies that $D \subset H$, since D is a subgraph of H, but H is disconnected.

To show that D is a component of H, it remains to show that D is not a proper subgraph of another connected subgraph of H. Suppose on the contrary that $D \subset \hat{D} \subseteq H$ and $\hat{D}$ is connected. Let $\hat{C}$ be the image of $\hat{D}$ under the isomorphism $(\varphi^{-1}|_{V(\hat{D})}, \psi^{-1}|_{E(\hat{D})})$, so $\hat{C} \simeq \hat{D}$ and $\hat{C}$ is connected by Proposition 10.74. Observing that $V(\hat{C}) = \varphi^{-1}(V(\hat{D}))$ and $E(\hat{C}) = \psi^{-1}(E(\hat{D}))$, we have

$$\begin{aligned}
D \subset \hat{D} &\Rightarrow [V(D) \subset V(\hat{D})] \vee [E(D) \subset E(\hat{D})] \\
&\Rightarrow [\varphi(V(C)) \subset V(\hat{D})] \vee [\psi(E(C)) \subset E(\hat{D})] \\
&\Rightarrow [V(C) \subset \varphi^{-1}(V(\hat{D}))] \vee [E(C) \subset \psi^{-1}(E(\hat{D}))] \\
&\Rightarrow [V(C) \subset V(\hat{C})] \vee [E(C) \subset E(\hat{C})] \\
&\Rightarrow C \subset \hat{C},
\end{aligned}$$

and hence C is a proper subgraph of another connected subgraph of G. As this is a contradiction, we conclude that D is a component of H. Since C is an arbitary component of G, it follows that each component C of G has a corresponding component D of H that is the image of C under the isomorphism (φ, ψ). The components of G are disjoint by Proposition 10.73, as are the components of H, and because $\varphi : V(G) \to V(H)$ and $\psi : E(G) \to E(H)$ are bijections, the correspondence between the components of G and the components of H is itself a bijection. Therefore G and H have the same number of components. ■

For the next definition, recall that for a graph G with $v \in V(G)$, the symbol $G - v$ denotes the subgraph of G induced by $V(G) - \{c\}$. This subgraph obtains from G by deleting v and all edges in $E(G)$ incident to v. In contrast, if $e \in E(G)$, then $G - e$ is the subgraph of G obtained by deleting the edge e.

Definition 10.76. *Let G be an undirected graph. We say $v \in V(G)$ is a* ***cut-vertex*** *of G if the graph $G - v$ has more connected components than G. We say $e \in E(G)$ is a* ***cut-edge*** *(or* ***bridge****) of G if $G - e$ has more connected components than G.*

Proposition 10.77. *Let G be a connected simple graph. Then $c \in V(G)$ is a cut-vertex of G if and only if there exist vertices $u \neq v$ in $V(G) - \{c\}$ such that every u, v-walk in G passes through c.*

Proof. Suppose for every $u \neq v$ in $V(G) - \{c\}$ there exists a u, v-walk in G that does not pass through c. Let u and v be arbitary distinct vertices of $G - c$. Since $u, v \in V(G) - \{c\}$, there is a u, v-walk W in G that does not pass through c; and since W does not pass through c, it does not traverse any edge $e \in E(G)$ that is incident to c. Hence all the vertices and edges of W belong to the subgraph $G - c$, which implies that W is a u, v-walk in $G - c$. Since any two distinct vertices in $G - c$ are connected, we conclude that $G - c$ is a connected graph, and therefore c is not a cut-vertex of G.

For the converse, suppose there exist vertices $u \neq v$ in $V(G) - \{c\}$ such that every u, v-walk in G passes through c. Then there is no walk that connects u and v in the subgraph $G - c$, implying that $G - c$ is disconnected, and therefore c is a cut-vertex of G. ■

For the statement of the next proposition we say an edge e in a graph G is a **cycle edge** if e is in the edge set of some cycle in G.

Proposition 10.78. *An edge of a graph is a cut-edge if and only if it is not a cycle edge.*

Proof. Let G be a graph and suppose e is a cycle edge in a connected component H of G. If e is a loop or there is an edge parallel to e, then $H - e$ is connected and e is not a cut-edge. We henceforth assume e is neither a loop nor an element of a multiedge. Fix $u, v \in V(H)$ with $u \neq v$. Then there is a u, v-walk W in H. If W does not traverse e then certainly W is a u, v-walk in $H - e$, and so assume that W traverses e. We have $W = u_1u_2 \cdots u_n$ for some $n \geq 2$, with $u = u_1$, $v = u_n$, and $e = u_ku_{k+1}$ for some $1 \leq k \leq n - 1$. Also e is an edge in a cycle $C = c_1c_2 \cdots c_mc_1$, so that $e = c_jc_{j+1}$ for some $1 \leq j \leq m$ (we define $c_{m+1} = c_1$). The direction of travel on C we choose to be such that $c_j = u_k$ and $c_{j+1} = u_{k+1}$. We now observe that the walk with vertex sequence $u_1u_2 \cdots u_{k-1}c_jc_{j-1} \cdots c_1c_m \cdots c_{j+1}u_{k+2}u_{k+3} \cdots u_n$ is a u, v-walk in $H - e$. We conclude that $H - e$ is connected, and therefore e is not a cut-edge.

Now suppose that edge e in component H is not a cut-edge. Certainly e is a cycle edge if it is a loop or an element of a multiedge, as we assume e is neither of these. We have $e = uv$ for distinct $u, v \in V(H)$. Now, since $H - e$ is connected, there is a u, v-walk $W = u_1u_2 \cdots u_n$ with

$u = u_1$ and $v = u_n$ in $H - e$. Since $e = u_n u_1$, we find that $u_1 u_2 \cdots u_n u_1$ is a cycle in H, and therefore e is a cycle edge. ■

Proposition 10.79. *Suppose v is an endpoint of a cut-edge e of the graph G, and there is no loop at v. Then v is a cut-vertex of G if and only if $d_G(v) \neq 1$.*

Proof. We shall start by assuming that $G = (V, E)$ is connected. Let ι be the incidence function for G. Since a cut-edge cannot be a loop, $\iota(e) = \{u, v\}$ for some $u \neq v$. By the definition of cut-edge, $G - e$ has two connected components: C_1 containing v and C_2 containing u.

Suppose $d_G(v) = 1$. Then e is the only edge incident to v, so that v is isolated in $G - e$, and we have $C_1 = (\{v\}, \varnothing)$ and $C_2 = (V - \{v\}, E - \{e\})$. In particular we see that $G - v$ is connected since $G - v = C_2$, and hence v is not a cut-vertex of G.

For the converse, suppose $d_G(v) \neq 1$. Then $d_G(v) \geq 2$ since G is connected, and so there is some $f \in E$ such that $e \neq f$ and f is incident to v. We observe that $\iota(f) \neq \{u, v\}$, since otherwise e and f would be parallel edges, and then e could not possibly be a cut-edge of G. Indeed, because there is no loop at v, we find that $\iota(f) = \{v, w\}$ for some $w \neq u, v$. Again $G - e$ has two components, with $C_2 = (V - \{v\}, E - \{e\})$ as before, but now C_1 contains the walk vfw. We now delete v from $G - e$ to obtain $G - v$, and observe that $G - v$ has two components: C_2, and a subgraph of C_1 that contains at least the vertex w. Therefore v is a cut-vertex of G.

Finally, if G is not connected, then we need only carry out the proof above on the component C of G that contains the edge e, so that $e \in E(C)$ and $u, v \in V(C)$. In this case e is a cut-edge of C. ■

The next proposition states that a connected undirected graph G consisting of n vertices must have at least $n - 1$ edges.

Proposition 10.80. *If G is a connected graph and $|V(G)| = n$, then $|E(G)| \geq n - 1$.*

Proof. Let $P(n)$ be the statement "If G is a loopless connected graph and $|V(G)| = n$, then $|E(G)| \geq n - 1$." We shall use induction to prove $P(n)$ for all $n \in \mathbb{Z}^+$, then prove the statement of the proposition.

To start, we observe that $P(1)$ is trivially true. Suppose $P(n)$ to be true for some $n \geq 1$, and let G be a loopless connected graph with $|V(G)| = n + 1$. We must show that $|E(G)| \geq n$, but by way of contradiction suppose that $|E(G)| \leq n - 1$. For $V(G) = \{v_1, \ldots, v_{n+1}\}$, the Handshaking theorem (Theorem 10.21) implies that

$$\sum_{i=1}^{n+1} d_G(v_i) = 2|E(G)| \leq 2(n - 1).$$

This inequality cannot be satisfied if $d(v_i) \geq 2$ for all $1 \leq i \leq n + 1$, so there exists some $v \in V(G)$ such that $d_G(v) \leq 1$. Indeed $d_G(v) = 1$ since G is connected. Remove this vertex v and its single incident edge e to obtain the subgraph $G - v$. Clearly e is a cut-edge: deleting it isolates v and thereby disconnects G; and since there is no loop at v, Proposition 10.79 implies that v is not a cut-edge, and hence $G - v$ is a loopless connected graph with $|V(G - v)| = n$ and $|E(G - v)| \leq n - 2$. However, this contradicts our inductive hypothesis $P(n)$, so that $|E(G)| \geq n$ must be the case, and we have proven $P(n + 1)$. Therefore $P(n)$ is true for all $n \in \mathbb{Z}^+$.

Now suppose that G is a connected graph with $|V(G)| = n$. Let $E'(G)$ consist of the edges in $E(G)$ that are not loops, so that $H = (V(G), E'(G))$ is a subgraph of G that is itself a loopless connected graph consisting of n vertices. Then $|E'(G)| \geq n - 1$ by $P(n)$, and since $|E(G)| \geq |E'(G)|$ the conclusion of the proposition follows. ■

Corollary 10.81. *If G is a graph with n vertices and k components, then $|E(G)| \geq n - k$.*

Proof. Suppose G is a graph with n vertices and components $G_1, \ldots, G_k$. For each $1 \leq i \leq k$ let $n_i = |V(G_i)|$. Then $|E(G_i)| \geq n_i - 1$ by Proposition 10.80, and hence

$$|E(G)| = \sum_{i=1}^{k} |E(G_i)| \geq \sum_{i=1}^{k} (n_i - 1) = \sum_{i=1}^{k} n_i - k = n - k$$

as claimed. ■

Proposition 10.80 does not assume G is a simple graph. The next proposition furnishes some results concerning the property of connectedness as it relates to the number of edges in a simple undirected graph.

Proposition 10.82. *Let G be a simple undirected graph.*

1. *If G has k connected components, with the ith component having n_i vertices for $1 \leq i \leq k$, then*
$$|E(G)| \leq \sum_{i=1}^{k} C(n_i, 2). \tag{10.7}$$
2. *If G has n vertices and k connected components, then*
$$|E(G)| \leq \frac{(n-k)(n-k+1)}{2}.$$
3. *If G has n vertices and*
$$|E(G)| > \frac{(n-1)(n-2)}{2},$$
then G is connected.

Proof.
Proof of (1). Suppose G has components $C_1, \ldots, C_k$ such that $|V(C_i)| = n_i$. In a simple graph with n vertices the greatest possible number of edges is realized when every vertex is joined to every other vertex to form K_n, the complete graph on n vertices. Thus the number of edges in G is maximal if and only if its ith component is the complete graph on n_i vertices for each $i \in \{1, \ldots, k\}$.[7] Since $|E(K_{n_i})| = C(n_i, 2)$ by Proposition 10.7, the inequality (10.7) follows.

Proof of (2). Suppose G has n vertices and components $C_1, \ldots, C_k$. Let $|V(C_i)| = n_i$ for each $i \in \{1, \ldots, k\}$, so that $n = n_1 + \cdots + n_k$. From part (1) we have

$$|E(G)| \leq \sum_{i=1}^{k} \frac{n_i!}{2!(n_i - 2)!} = \frac{1}{2} \sum_{i=1}^{k} n_i(n_i - 1) = \frac{1}{2}\left(\sum_{i=1}^{k} n_i^2 - n\right),$$

[7] Strictly speaking we require that $C_i \simeq K_{n_i}$.

so to complete the proof it will be sufficient to show that, for all $k \geq 1$, if $n = \sum_{i=1}^{k} n_i$ for $n_1, \dots, n_k \in \mathbb{Z}^+$, then

$$\sum_{i=1}^{k} n_i^2 \leq (n-k)(n-k+1) + n. \tag{10.8}$$

We accomplish this by induction on k. For the basis step we set $k = 1$, and note that (10.8) then reduces to the verity $n_1^2 \leq n_1^2$.

For the inductive step, fix $k \geq 1$ and assume that (10.8) holds whenever $n = \sum_{i=1}^{k} n_i$ for $n_1, \dots, n_k \in \mathbb{Z}^+$. Now suppose $n = \sum_{i=1}^{k+1} n_i$ for $n_1, \dots, n_{k+1} \in \mathbb{Z}^+$. We must show that

$$\sum_{i=1}^{k+1} n_i^2 \leq (n-k-1)(n-k) + n. \tag{10.9}$$

Since $\sum_{i=1}^{k} n_i = n - n_{k+1}$, by the inductive hypothesis we have

$$\sum_{i=1}^{k} n_i^2 \leq [(n - n_{k+1}) - k][(n - n_{k+1}) - k + 1) + (n - n_{k+1}),$$

and so, adding n_{k+1}^2 to both sides and substituting $\sum_{i=1}^{k} n_i$ for $n - n_{k+1}$,

$$\sum_{i=1}^{k+1} n_i^2 \leq n_{k+1}^2 + \left(\sum_{i=1}^{k} n_i - k\right)\left(\sum_{i=1}^{k} n_i - k + 1\right) + \sum_{i=1}^{k} n_i.$$

To obtain (10.9) we thus need to show that

$$n_{k+1}^2 + \left(\sum_{i=1}^{k} n_i - k\right)\left(\sum_{i=1}^{k} n_i - k + 1\right) + \sum_{i=1}^{k} n_i \leq \left(\sum_{i=1}^{k+1} n_i - k - 1\right)\left(\sum_{i=1}^{k+1} n_i - k\right) + \sum_{i=1}^{k+1} n_i.$$

This is accomplished by expanding the products on each side of the inequality and combining like terms to obtain the equivalent inequality

$$(n_{k+1} - 1)\left(\sum_{i=1}^{k} n_i - k\right) \geq 0.$$

This inequality is true since $n_i \geq 1$ for all $1 \leq i \leq k+1$, and so the proof is done.

Proof of (3). The statement is vacuously true when $n = 1$, so suppose G has $n \geq 2$ vertices. Suppose further that G is disconnected. Then G has $k \geq 2$ connected components, and so $|E(G)| \leq \frac{1}{2}(n-k)(n-k+1)$ by part (2). Now, $k \leq n$ since each component of G must contain at least one vertex, and so, observing that

$$\frac{(n-k)(n-k+1)}{2} \leq \frac{(n-2)(n-1)}{2}$$

for each $2 \leq k \leq n$, we conclude that $|E(G)| \leq \frac{1}{2}(n-2)(n-1)$. Therefore $|E(G)| > \frac{1}{2}(n-1)(n-2)$ if G is connected. ∎

The notion of connectedness is more nuanced when applied to digraphs, with there being two senses in which a digraph may be said to be "connected." In order to define the "weaker" sense of connectedness, we need the notion of the "underlying graph" of a digraph.

Definition 10.83. *Let D be a digraph, and for each $e \in E(D)$ let $\xi(e)$ be an undirected edge whose endpoints coincide with the head and tail of e. The* ***underlying graph*** *of D is the undirected graph G for which $V(G) = V(D)$ and $E(G) = \{\xi(e) : e \in E(D)\}$.*

We see in the definition that ξ is a function on $E(D)$ that strips away a directed edge's orientation and thereby transforms it into an undirected edge. Thus the underlying graph of a digraph is the undirected graph that results when all the arrows in the digraph's drawing have been removed.

Recalling that vertices u and v are said to be mutually reachable in a digraph D if there exists both a directed u, v-walk and a directed v, u-walk in D, we make the following definition.

Definition 10.84. *Let D be a digraph. We say D is* ***weakly connected*** *if its underlying graph is connected, and* ***strongly connected*** *if every pair of vertices $u, v \in V(D)$ are mutually reachable.*

A strongly connected digraph D is also weakly connected, since the mutual reachability of any two vertices $u, v \in V(D)$ implies there exists an undirected u, v-walk in the underlying graph of D, and hence u and v are connected as defined in Definition 10.71.

10.6 – Vertex and Edge Connectivity

The **connectivity** of a connected graph G is some measure of the degree to which G is connected, an inquiry that typically begins by asking what is the minimum number of vertices or edges that must be removed from G in order to disconnect the graph.

Definition 10.85. *Let G be an undirected graph. A **vertex cut** of G is any set $S \subseteq V(G)$ such that the graph $G - S$ is either disconnected or has only one vertex. If $\dot{\mathcal{K}}(G)$ is the collection of all vertex cuts of G, then the **vertex connectivity** of G is*

$$\dot{\kappa}(G) = \min\{|S| : S \in \dot{\mathcal{K}}(G)\},$$

*with any $S \in \dot{\mathcal{K}}(G)$ such that $|S| = \dot{\kappa}(G)$ being a **minimal vertex cut**. We say G is k-**vertex-connected** (or k-**connected**) if $\dot{\kappa}(G) \geq k$.*

Certainly $G - \varnothing$ is disconnected if G is disconnected, in which case $\varnothing \in \dot{\mathcal{K}}(G)$ and hence $\dot{\kappa}(G) = 0$. If G is a complete graph, so that $G \simeq K_n$ for some $n \geq 1$, then $G - S$ is connected for any $S \subseteq V(G)$, and hence the only possible vertex cut is one in which all but one vertex in $V(G)$ is removed from the graph. Therefore $\dot{\kappa}(G) = n - 1$ whenever $G \simeq K_n$. The reader should verify that $\dot{\kappa}(G) \leq |V(G)| - 2$ whenever G is a noncomplete graph.

Definition 10.86. *Let G be an undirected graph. An **edge cut** of G is any set $S \subseteq E(G)$ such that the graph $G - S$ is disconnected. If $\bar{\mathcal{K}}(G)$ is the collection of all edge cuts of G, then the **edge connectivity** of G is*

$$\bar{\kappa}(G) = \min\{|S| : S \in \bar{\mathcal{K}}(G)\},$$

*with any $S \in \bar{\mathcal{K}}(G)$ such that $|S| = \bar{\kappa}(G)$ being a **minimal edge cut**. We say G is k-**edge-connected** if $\bar{\kappa}(G) \geq k$.*

If G is disconnected, then $\varnothing \subseteq E(G)$ is such that $G - \varnothing$ is disconnected, and so $\bar{\kappa}(G) = 0$ since $\varnothing \in \bar{\mathcal{K}}(G)$. It is also the case that $\bar{\kappa}(G) = n - 1$ whenver $G \simeq K_n$, since each vertex in K_n is joined by precisely $n-1$ edges to the other vertices in the graph. Thus $\dot{\kappa}(K_n) = \bar{\kappa}(K_n) = n-1$.

The dot and bar decorations in the symbols $\dot{\kappa}(G)$ and $\bar{\kappa}(G)$ are not standard notations, but because vertices are frequenty denoted by dots and edges by line segments, it should be easy to remember which symbol pertains to vertex connectivity and which to edge connectivity.[8]

Whilst on the subject of convenient notations, we define one more that does in fact pervade the literature: given an undirected graph G and sets $S, T \subseteq V(G)$, define $[S, T] \subseteq E(G)$ to be the set of edges in $E(G)$ with one endpoint in S and the other in T. The reader may verify that $[S, S^c]$ is an edge cut for G if, and only if, $\varnothing \neq S \subset V(G)$ and $S^c = V(G) - S$.

Lemma 10.87. *Let G be a simple noncomplete connected graph and $\varnothing \neq S \subset V(G)$. If $[S, S^c]$ is an edge cut for G such that every vertex in S is adjacent to every vertex in S^c, then $|[S, S^c]| \geq |V(G)| - 1$.*

---

[8]Many authors use κ and λ where we use $\dot{\kappa}$ and $\bar{\kappa}$, respectively. The use of the Greek letter κ (kappa), which corresponds to k in the Latin alphabet, is motivated by the k sound in the word “cut.”

Proof. Let $n = |V(G)|$ and $k = |S|$. Then $1 \leq k \leq n-1$ since $\varnothing \neq S \subset V(G)$, so that $(k-1)(n-k-1) \geq 0$, and hence $k(n-k) \geq n-1$. Also $|[S, S^c]| = |S||S^c|$ since each $v \in S$ is joined by $|S^c|$ edges to each vertex in S^c. Observing that $|S^c| = n-k$, we now have

$$\big|[S, S^c]\big| = |S||S^c| = k(n-k) \geq n-1 = |V(G)|-1,$$

finishing the proof. ■

For the statement of the next theorem we define the **minimal degree** of an undirected graph G, denoted by $\delta(G)$, to be the smallest degree attained by the vertices of G; that is,

$$\delta(G) = \min\{d_G(v) : v \in V(G)\}.$$

Thus if G has an isolated vertex, then $\delta(G) = 0$.

Theorem 10.88. *Let G be a simple connected graph. Then $\dot{\kappa}(G) \leq \bar{\kappa}(G) \leq \delta(G)$.*

Proof. Let $v \in V(G)$ be such that $d_G(v) = \delta(G)$. Then $\delta(G)$ edges $e_1, \ldots, e_{\delta(G)}$ are incident to v, so that $F = \{e_1, \ldots, e_{\delta(G)}\}$ is an edge cut for G. Since $\bar{\kappa}$ is the cardinality of a minimal edge cut, we have $\bar{\kappa}(G) \leq |F| = \delta(G)$.

If G is complete, so that $G \simeq K_n$ for some $n \geq 1$, then each of the n vertices of G is joined by a single edge to each of the other $n-1$ vertices, and so the minimum number of edges required to disconnect G is $n-1$; that is, $\bar{\kappa}(G) = n-1$. Meanwhile $\dot{\kappa}(G) = n-1$ by definition in this case, and we conclude that $\dot{\kappa}(G) \leq \bar{\kappa}(G)$.

We now show that $\dot{\kappa}(G) \leq \bar{\kappa}(G)$ under the assumption that G is noncomplete. For a minimal edge cut $[S, S^c]$, suppose every vertex in S is adjacent to every vertex in S^c. Then Lemma 10.87 implies that

$$\bar{\kappa}(G) = |[S, S^c]| \geq |V(G)| - 1 > \dot{\kappa}(G).$$

The only other possibility is that at least one vertex $u \in S$ is not adjacent to at least one vertex $v \in S^c$. Let C_1 be the set of neighbors of u in S^c, and let C_2 be the set of vertices in $S - \{u\}$ having at least one neighbor in S^c. Thus

$$C_1 = N_G(u) \cap S^c \quad \text{and} \quad C_2 = \big\{x \in S - \{u\} : N_G(x) \cap S^c \neq \varnothing\big\}.$$

Let $C = C_1 \cup C_2$. Since $E(G - C_1)$ lacks those edges in $E(G)$ that join $u \in S$ to any $y \in S^c$, and $E(G - C_2)$ lacks those edges in $E(G)$ that join any $x \in S - \{u\}$ to any $y \in S^c$, the graph $G - C$ possesses no edges that join any vertex in $S - C_2$ (which at least contains u) to any vertex in S^c (which at least contains v). Thus $G - C$ has connected components $S - C_2$ and $S^c - C_1$, and we find that C is a vertex cut for G. Moreover, each $x \in C_1$ corresponds to a unique edge $\{u, x\} \in [S, S^c]$, and each $x \in C_2$ (so $x \neq u$) can be made to correspond to a unique edge $\{x, y\} \in [S, S^c]$ by choosing some particular $y \in N_G(x) \cap S^c$. In this way we obtain a one-to-one correspondence between the vertices of C and some subset of $[S, S^c]$, and therefore $\bar{\kappa}(G) = |[S, S^c]| \geq |C| \geq \dot{\kappa}(G)$. This finishes the proof. ■

In the proof of Theorem 10.88 we in fact found that $\dot{\kappa}(G) = \bar{\kappa}(G)$ in the case when G is complete. The converse is also true and is left as an exercise for the reader: If $\dot{\kappa}(G) = \bar{\kappa}(G)$, then G is complete.

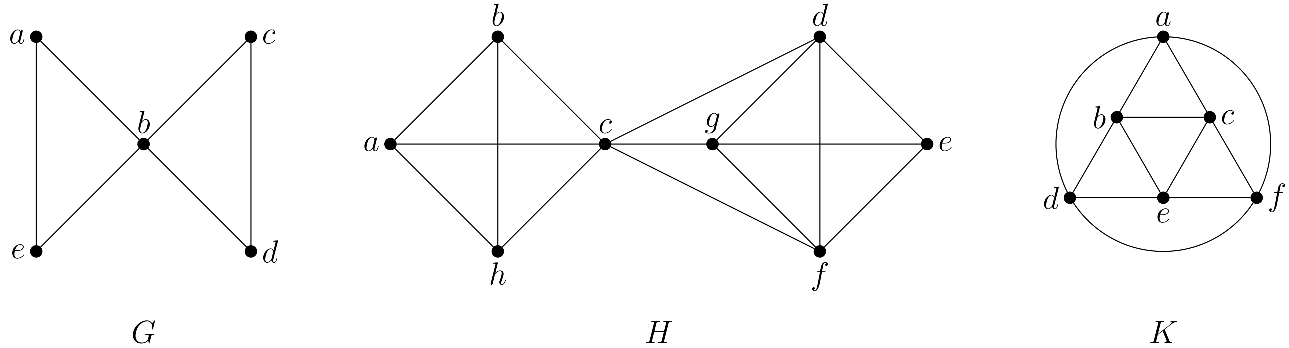


Figure 15.

Example 10.89. For the graph G in Figure 15, the set $\{b\} \subseteq V(G)$ is a vertex cut, and hence b itself is a cut-vertex. Since there is no smaller subset of $V(G)$ that is a vertex cut for G, we conclude that $\dot{\kappa}(G) = 1$. To determine the edge connectivity of G, we first observe that the deletion of any one edge from the graph does not disconnect it; however, deletion of edges ab and be (or bc and bd) does disconnect the graph, so that $S = \{ab, be\}$ is an edge cut, and therefore $\bar{\kappa}(G) = 2$. By inspection it's seen that $\delta(G) = 2$, and so in this case $\dot{\kappa}(G) < \bar{\kappa}(G) = \delta(G)$.

Now we consider graph H in Figure 15. There is one cut-vertex, namely c, so that $\{c\}$ is a vertex cut and hence $\dot{\kappa}(H) = 1$. Turning to the question of edge connectivity, we find that deleting any one of the 15 edges does not disconnect H, nor does deleting any two edges. Because $\delta(H) = 3$ and we must have $\bar{\kappa}(H) \leq \delta(H)$ by Theorem 10.88, it becomes clear that $\bar{\kappa}(H) = 3$. In particular $S = \{bc, ac, ch\}$ is an edge cut of cardinality 3. Once again we have $\dot{\kappa}(H) < \bar{\kappa}(H) = \delta(H)$.

Finally we cast our attention toward graph K in Figure 15. Deletion of any one of the six vertices does not disconnect K, nor does deleting any two of the vertices. Deleting a, d, and f leaves the cycle bce, which of course is a connected graph. Similarly, deleting b, c, and e leaves the cycle adf. Indeed, deleting any three vertices does not disconnect K, so we must have $\dot{\kappa}(K) \geq 4$. However, we also see that $\delta(K) = 4$, so $4 \leq \dot{\kappa}(K) \leq \bar{\kappa}(K) \leq \delta(K) = 4$, and therefore $\dot{\kappa}(K) = \bar{\kappa}(K) = \delta(K) = 4$. ■

Example 10.90. Here we investigate the vertex and edge connectivity of complete bipartite graphs $K_{m,n}$ for $m, n \in \mathbb{Z}^+$, which are always connected. First we consider $K_{1,5}$, shown at left in Figure 16. Deleting u disconnects the graph, so that $\dot{\kappa}(K_{1,5}) = 1$. Also, deleting any one edge disconnects the graph, so $\bar{\kappa}(K_{1,5}) = 1$. We have $\dot{\kappa}(K_{1,5}) = \bar{\kappa}(K_{1,5}) = 1$.

Now we examine $K_{2,3}$, shown at right in Figure 16. Deleting any single vertex results in a graph that is still connected. Indeed, deleting either u_1 or u_2 results in the graph $K_{1,3}$, while deleting one of the vertices v_1, v_2, or v_3 yields the graph $K_{2,2}$. Deleting both u_1 and u_2 does disconnect the graph, however, so $\dot{\kappa}(K_{2,4}) = 2$. As for the edge connectivity, deleting any single edge fails to disconnect the graph, but deleting edges u_1v_1 and u_2v_1 does disconnect it. Therefore $\dot{\kappa}(K_{2,4}) = \bar{\kappa}(K_{2,4}) = 2$.

Finally we turn to $K_{m,n}$, where $V(K_{m,n})$ has a bipartition (U, V) with $U = \{u_1, \ldots, u_m\}$ and $V = \{v_1, \ldots, v_n\}$. Without loss of generality we assume that $1 \leq m \leq n$. Deleting a single vertex in U yields the graph $K_{m-1,n}$, while deleting a single vertex in V yields $K_{m,n-1}$. From these observations it is clear that deleting $1 \leq i \leq m$ vertices from U and $1 \leq j \leq n$ vertices

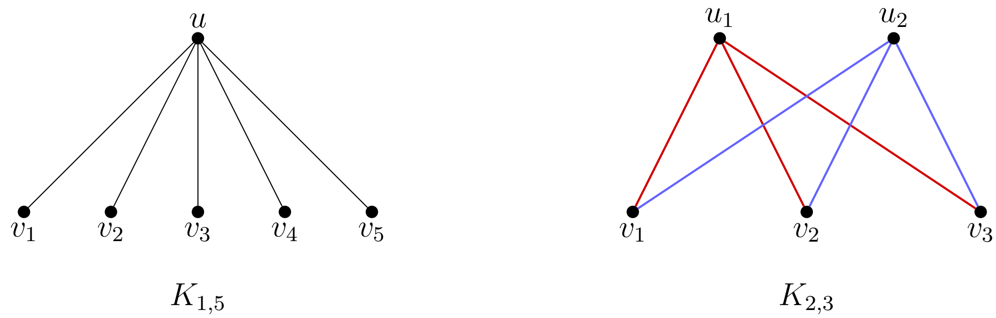


FIGURE 16.

from V results in the graph $K_{m-i,n-j}$, and so the least number of vertices that must be deleted in order to disconnect the graph is m (with the set U being the associated minimal vertex cut). Hence $\dot{\kappa}(K_{m,n}) = m$, and since $\delta(K_{m,n}) = m$ and $\dot{\kappa}(K_{m,n}) \leq \bar{\kappa}(K_{m,n}) \leq \delta(K_{m,n})$, it is clear that $\dot{\kappa}(K_{m,n}) = \bar{\kappa}(K_{m,n}) = \delta(K_{m,n}) = m$ if $m \leq n$. ∎

10.7 – Eulerian and Hamiltonian Walks

In this section we consider two kinds of walks in a graph G, whether directed or undirected: a kind that traverses every edge of G exactly once, and a kind that passes through every vertex of G exactly once. Before wading in any further we need to introduce some terminology.

Definition 10.91. *Let G be an undirected graph or digraph. An **Eulerian trail** in G is a trail that traverses every edge of G, and an **Eulerian circuit** in G is a circuit that traverses every edge of G. A graph is **Eulerian** if it contains an Eulerian circuit.*

*A **Hamiltonian path** in G is a path that passes through every vertex of G, and a **Hamiltonian cycle** in G is a cycle that passes through every vertex of G. A graph is **Hamiltonian** if it contains a Hamiltonian cycle.*

If G is a trivial graph, so that it possesses only one vertex v and no edges, then the trivial walk v is considered to be both an Eulerian circuit and a Hamiltonian cycle in G. Thus any trivial graph is both Eulerian and Hamiltonian, as is any graph consisting of two or more vertices and no edges.

Eulerian trails and circuits we shall collectively call **Eulerian walks**, while Hamiltonian paths and cycles are **Hamiltonian walks**. Being a trail, an Eulerian trail in G must traverse every edge of G *exactly once*, which necessarily implies that it must pass through every vertex of G at least once. An Eulerian circuit is simply a closed Eulerian trail. A Hamiltonian path in G, being a path, must pass through every vertex of G *exactly once*, but need not traverse every edge (consider a graph with parallel edges or a loop). A Hamiltonian cycle passes through every vertex of G exactly once, except that it ends at the same vertex where it starts.

Though the definitions we've furnished thus far in this section apply to both undirected graphs and digraphs, we shall henceforth restrict our attention to the former in all theoretical developments and examples. We consider Eulerian walks first (with special attention given to circuits), then turn to Hamiltonian walks (emphasizing cycles).

The following lemma will be needed for part of the proof of our first theorem concerning Eulerian graphs.

Lemma 10.92. *Let G be an undirected graph. If $d_G(v) \geq 2$ for all $v \in V(G)$, then G contains a cycle.*

Proof. Suppose every vertex of G has degree at least 2. Any loop in G is a cycle, and any parallel edges in G also form a cycle, so we assume henceforth that G is a simple graph. As usual we take G to be finite, so $|V(G)|$ is an upper bound on the set of all possible path lengths in G. Let P be a path of maximum possible length, with starting vertex v_1. Since $d_G(v_1) \geq 2$ and G is simple, v_1 is adjacent to some vertex $v_2 \neq v_1$; and since $d_G(v_2) \geq 2$, v_2 must be adjacent to some vertex $v_3 \neq v_1$. Hence P has vertex sequence $v_1 v_2 \cdots v_n$ for some $n \geq 3$. Now, $d_G(v_n) \geq 2$ implies that v_n is adjacent to at least one other vertex besides v_{n-1}, but all the neighbors of v_n must already be contained in P. Indeed, if this were not the case, then P could be extended to an $(n+1)$st vertex, which contradicts the hypothesis that P has maximum length. Thus, besides v_{n-1}, the vertex v_n must have a neighbor v_k for some $1 \leq k \leq n-2$, and therefore $C = v_k v_{k+1} \cdots v_n v_k$ is a cycle in G. ∎

It will be convenient to define a graph G to be **even** if $d_G(v)$ is even for all $v \in V(G)$, and **odd** if $d_G(v)$ is odd for all $v \in V(G)$. Any graph with at least one vertex and no edges is even.

Theorem 10.93. *Let G be a nonempty undirected graph. Then G is Eulerian if, and only if, G is even and has at most one nontrivial connected component.*

Proof. We first consider the case when G is a trivial graph with vertex v. Then G is Eulerian, has no nontrivial connected component, and $d_G(v) = 0$ (an even number), and so the statement of the theorem is trivially true. Henceforth we assume that G possesses at least one edge.

Suppose G is Eulerian, and for $n \geq 1$ let $C = v_0 e_1 v_1 \cdots e_n v_n$ with $v_0 = v_n$ be an Eulerian circuit. We observe that C is a nontrivial connected subgraph of G: any two distinct vertices v_i and v_j in C, where $i < j$, are connected by a walk of the form $v_i e_{i+1} v_{i+1} \cdots e_j v_j$, and therefore C must be contained in some nontrivial connected component H of G. Since C traverses every edge in G, it follows that H must contain all the edges in G, and hence any other connected component that G may possess must have no edges and so be trivial. This proves that G has at most one nontrivial connected component.

We now show that $d_G(v)$ is even for all $v \in V(G)$. Any isolated vertex of G has degree 0, and so it remains to reckon with the vertices in the nontrivial connected component H of G (which exists since we are assuming $E(G) \neq \varnothing$). First, every $v \in V(H)$ must have an incident edge, since otherwise v would constitute a trivial component of G and so not be contained in H. Then, since the Eulerian circuit C traverses every edge in H, and hence passes through every endpoint of every edge, it follows that C passes through every $v \in V(H)$. Let v be an arbitrary vertex in H, let e be incident to v, and commence to travel through the vertex-edge sequence of C starting at e. Since C passes through every edge in G precisely once, we'll know we've traveled the circuit precisely once as soon as we arrive back at e. Now, each time we encounter v while navigating the circuit, we must either enter and exit v by two distinct edges we've not traversed before, or else traverse a loop at v. Thus the edges incident to v consist of enter/exit pairs and loops. Each enter/exit pair and each loop contribute 2 to the degree of v, and therefore $d_G(v)$ is even.

The converse we prove using strong induction on the number of edges in G. Let $P(n)$ be the statement "If G is even, has at most one nontrivial component, and $|E(G)| = n$, then G is Eulerian." If $|E(G)| = 0$, then G is a trivial graph or a disjoint union of trivial graphs, hence Eulerian, and therefore $P(0)$ is true. Now fix $n \geq 0$, and suppose $P(k)$ is true for all $0 \leq k \leq n$. Let G be an even graph with at most one nontrivial component and $n+1$ edges. Then G has precisely one nontrivial component H containing all the edges, and possibly one or more trivial components that need not concern us. The connected graph H has no isolated points and must itself be even, so that $d_H(v) \geq 2$ for all $v \in V(H)$, and hence H contains a cycle C by Lemma 10.92. If $H = C$, then C is an Eulerian circuit in G and we're done; otherwise, since C must have at least one edge, deleting its edges from H yields a new graph H' with $V(H') = V(H)$ and $0 < |E(H')| \leq n$ (we note that $E(H') = \varnothing$ implies $H = C$). The graph H' is also even, since $d_{H'}(v) = d_H(v)$ if $v \notin V(C)$, and $d_{H'}(v) = d_H(v) - 2$ if $v \in V(C)$. It may be that H' is disconnected, but each nontrivial connected component of H' must be an even graph with at most n edges, and so is Eulerian by our inductive hypothesis. Any trivial component of H' must be a vertex $v \in V(C)$ such that $d_H(v) = 2$.

Let K be a nontrivial component of H', so there is no $e \in E(H')$ joining any $u \in V(K)$ to any $v \in V(H) - V(K)$. Suppose K contains no vertex in the cycle C, so that $V(C) \subseteq V(H) - V(K)$.

Then, since the edges in $E(H) - E(H')$ only join vertices in $V(C)$, there is no edge $e \in E(H)$ joining any $u \in V(K)$ to any $v \in V(H) - V(K)$, and hence H is disconnected—a contradiction. We conclude that each nontrivial component of H' must contain a vertex in C, in addition to being Eulerian.

Let C have vertex sequence $c_1 c_2 \cdots c_m c_1$, suppose H' has $\ell \geq 1$ nontrivial components, and for $1 \leq i_1 < \cdots < i_\ell \leq m$ let $c_{i_1}, \ldots, c_{i_\ell}$ be such that there exists a component H'_j of H' that contains vertex c_{i_j} of C . Let C_j be an Eulerian circuit in H'_j with starting vertex c_{i_j}, and in the graph H let W_j be the walk on C from c_{i_j} to $c_{i_{j+1}}$, designating $c_{i_{\ell+1}} := c_{i_1}$. Then $C_1 * W_1 * C_2 * W_2 * \cdots * C_\ell * W_\ell$ is a circuit in H containing all the edges in H, which contains all the edges in G, and therefore G is Eulerian. ■

Corollary 10.94. *Let G be a connected undirected graph with distinct vertices u and v. Then G has an Eulerian u,v-trail if, and only if, u and v have odd degree, and all $w \in V(G) - \{u, v\}$ have even degree.*

Proof. Suppose T is an Eulerian u,v-trail. Insert a new edge e with endpoints u and v into G. Then $T * veu$ is an Eulerian circuit in $G + e$, and by Theorem 10.93 the graph $G + e$ is even; that is, $d_{G+e}(w)$ is even for all $w \in V(G + e) = V(G)$. Now, $d_G(w) = d_{G+e}(w)$ for $w \in V(G) - \{u, v\}$, and $d_G(w) = d_{G+e}(w) - 1$ for $w \in \{u, v\}$, and therefore all $w \in V(G) - \{u, v\}$ have even degree while u and v have odd degree.

For the converse, suppose u and v have odd degree, and all $w \in V(G) - \{u, v\}$ have even degree. Again insert a new edge e into G that joins u and v in a new graph $G + e$. Now, $G + e$ is a connected graph whose vertices all have even degree, so by Theorem 10.93 there is an Eulerian circuit C in $G + e$. If we designate u to be the starting vertex of C, then $C = ue_1v_1 \cdots e_{n-1}v_{n-1}e_nu$ for some $n \geq 2$. For some $1 \leq k \leq n$ we have $e_k = e$, and thus $v_{k-1}e_kv_k$ is either veu or uev. By reversing the direction of travel along C if necessary, we can ensure that $v_{k-1}e_kv_k = veu$, and so in particular $v_{k-1} = v$ and $v_k = u$. Deleting e from C leaves us with trails $T_1 = ue_1v_1 \ldots e_{k-1}v$ and $T_2 = ue_{k+1}v_{k+1} \cdots e_nu$ in G, so that $T_2 * T_1$ is a u,v-trail in G that contains all the edges in G, and therefore G has an Eulerian u,v-trail. ■

Example 10.95. The origin of graph theory is by a broad consensus traced to a famous problem solved by the Swiss mathematician Leonhard Euler (pronounced *OI-ler*) in 1736. The town of Königsberg (now Kaliningrad), located in the Kingdom of Prussia, straddled the Pregel river and also included a couple of islands in the middle of the river. There were seven bridges that enabled the townsfolk to reach the islands and cross the river, as illustrated at left in Figure 17. Strolling across all seven bridges was a favorite pastime on Sundays, and the question arose whether it was possible to cross all seven bridges precisely once and return to the same place one started. Frederick the Great brought the problem to the attention of Euler, and Euler produced a proof that such an excursion is impossible.

To analyze the problem, the bridges can be represented by edges in a graph having four vertices: u and v to represent each island, and w and x to represent each bank of the river. What results is a multigraph like that shown at right in Figure 17. To ask whether it's possible to set off from any point in the town, cross all seven bridges precisely once, and return to the same point, is to ask whether the multigraph is Eulerian. And the answer is no, since the vertices in the graph do not all have even degree. Indeed, none of the vertices have even degree,

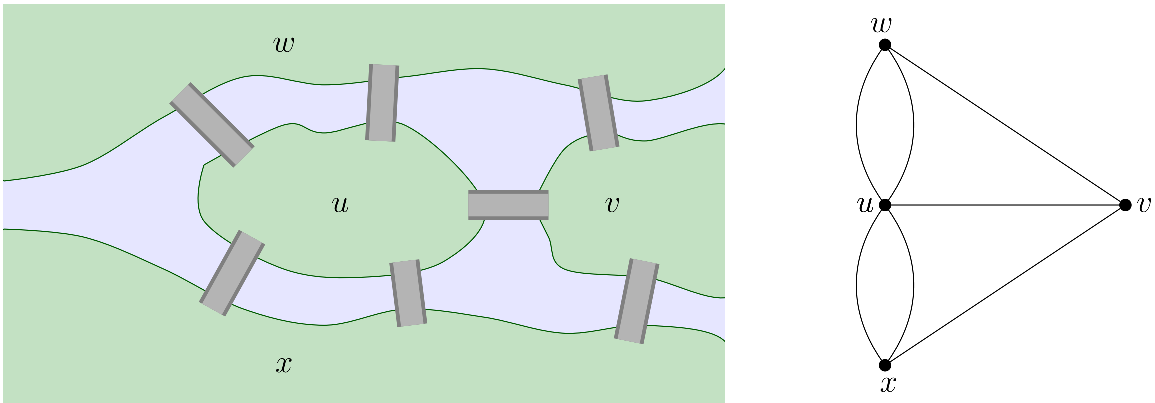


FIGURE 17. At left: the seven bridges of Königsberg; at right: a graph representation.

but Theorem 10.93 makes clear that it only takes one vertex with odd degree to ensure a graph is not Eulerian.

In fact, it is still not possible to cross all seven bridges precisely once even if we allow for an excursion that ends at a *different* point than where it began, such as starting at vertex u and ending at vertex v. For such an Eulerian u, v-trail to exist, Corollary 10.94 informs us that, in addition to u and v having odd degree (which is the case here), it is necessary for the other vertices w and x to have even degree (which is not the case). At least one bridge will need to be crossed twice, or not at all. ■

To date there is no known set of necessary and sufficient conditions for a graph G to be Hamiltonian; and so, while according to Theorem 10.93 a connected graph is Eulerian if and only if it is even, there is no theorem that states that a graph is Hamiltonian if and only if it satisfies one or more stated properties. Certainly any cycle C_n or complete graph K_n is Hamiltonian, but these are special classes of graphs. There are, however, theorems that give *sufficient* conditions for a graph to be Hamiltonian. One such theorem, established in 1960, is the following.

Theorem 10.96 (Ore's Theorem). *Let G be a simple undirected graph. If $|V(G)| \geq 3$ and $d_G(u) + d_G(v) \geq |V(G)|$ for all $u, v \in V(G)$ such that $\{u, v\} \notin E(G)$, then G is Hamiltonian.*

Proof. Suppose that G is not Hamiltonian. Let $n = |V(G)|$, and assign some order to the n vertices of G. Now we join the first vertex to as many other vertices in G as possible without introducing loops or parallel edges, or obtaining a Hamiltonian graph. We then perform the same operation on the second vertex, the third, and so on until the nth vertex is operated on. What results is a new simple graph H with $V(H) = V(G)$ and $E(H) \supseteq E(G)$ that is so constructed that the addition of a single new edge to H that is neither a loop nor parallel to an existing edge will result in a simple graph that is Hamiltonian. (That enough edges may be added to G to obtain a Hamiltonian graph is clear from the fact that the complete graph on n vertices, K_n, is Hamiltonian.) Let e be such an edge, so that $H + e$ is Hamiltonian and thus has a Hamiltonian cycle $C = u_1e_1 \cdots u_ne_nu_1$. That C traverses e is evident from the fact that, if it did not, then deleting e would leave C intact as a Hamiltonian cycle in H, which is impossible. Hence $e = e_k$ for some $1 \leq k \leq n$, and deleting e results in one of three outcomes in H: two paths $P_1 = u_1 \cdots u_{k-1}u_k$ and $P_2 = u_{k+1} \cdots u_n u_1$ (if $2 \leq k \leq n-1$), or a path $P_3 = u_2 \cdots u_n u_1$ (if $k = 1$), or a path $P_4 = u_1 u_2 \cdots u_n$ (if $k = n$). Certainly P_4 and P_3 are Hamiltonian paths in H, while in the case when $2 \leq k \leq n-1$ we find $P_2 * P_1$ to be a Hamiltonian path in H. Therefore H has a Hamiltonian path P, which with a relabeling of the vertices we may write as $P = v_1 v_2 \cdots v_n$. That H has no Hamiltonian cycle implies v_1 and v_n are not adjacent in H, hence not adjacent in G, and so

$$d_H(v_1) + d_H(v_n) \geq d_G(v_1) + d_G(v_n) \geq n,$$

whence $d_H(v_n) \geq |V(G)| - d_H(v_1)$, and therefore v_n is not adjacent to at most $d_H(v_1)$ vertices in H.

Next we define a set S consisting of all vertices $v_i \in V(H) = \{v_1, \ldots, v_n\}$ such that v_{i+1} is adjacent to v_1 in H:

$$S = \big\{v_i \in V(H) : \{v_1, v_{i+1}\} \in E(H)\big\}.$$

Clearly $v_n \notin S$, and since v_1 is not adjacent to itself or v_n, there is a one-to-one correspondence between the elements of S and the vertices adjacent to v_1 in H, so that $|S| = d_H(v_1)$. Now, if v_n were not adjacent to any $v \in S$, then because $v_n \notin S$ and v_n is not adjacent to itself, it would follow that v_n is not adjacent to $d_H(v_1) + 1$ vertices, thereby contradicting our earlier finding that v_n is not adjacent to at most $d_H(v_1)$ vertices. Hence S contains some v_ℓ such that $\{v_\ell, v_n\} \in E(H)$, where $\{v_1, v_{\ell+1}\} \in E(H)$ by the definition of S. With these two edges and the Hamiltonian path P, we may now construct a Hamiltonian cycle

$$C = v_1 v_2 \cdots v_\ell v_n v_{n-1} \cdots v_{\ell+1} v_1,$$

in contradiction to H being non-Hamiltonian. Therefore G must be Hamiltonian. ■

Ore's theorem is an improvement on another theorem, now called Dirac's theorem, that was estalished in 1952. Dirac's theorem is an immediate corollary to Ore's theorem.

Theorem 10.97 (Dirac's Theorem). *Let G be a simple undirected graph. If $|V(G)| \geq 3$ and $d_G(v) \geq \frac{1}{2}|V(G)|$ for all $v \in V(G)$, then G is Hamiltonian.*

Proof. Suppose $|V(G)| \geq 3$ and $d_G(v) \geq \frac{1}{2}|V(G)|$ for all $v \in V(G)$. Then for any nonadjacent $u, v \in V(G)$ we have $d_G(u) + d_G(v) \geq \frac{1}{2}|V(G)| + \frac{1}{2}|V(G)| = |V(G)|$, and therefore G is Hamiltonian by Ore's theorem. ■

Proposition 10.98. *For all $n \geq 3$, K_n is Hamiltonian.*

Proof. Let $n \geq 3$, and fix $v \in V(K_n)$. Then $d(v) = n - 1$ since v is joined to each of the other $n - 1$ vertices in the graph by a single edge. Now,

$$d(v) \geq \frac{1}{2}|V(K_n)| \quad \Leftrightarrow \quad n - 1 \geq \frac{n}{2} \quad \Leftrightarrow \quad n \geq 2,$$

and since $n \geq 2$ is true, so too is $d(v) \geq \frac{1}{2}|V(K_n)|$. Therefore K_n is Hamiltonian by Dirac's theorem. ■

10.8 – Weighted Graphs and Shortest Paths

The definition for the length of a walk in a graph G, given in §10.2, indicates that for any vertices $u, v \in V(G)$, the length of a u, v-path in G equals the number of edges that are traversed by the path, which equals the number of edges in the path's vertex-edge sequence. This definition effectively sets the length of any edge $e \in E(G)$ to be equal to 1. However, for many applications it makes sense to give each edge in a graph its own designated length, or "weight," which gives rise to what's called a weighted graph.

Definition 10.99. *A* ***weighted graph*** *is a graph G in which each edge e has an associated number, called the* ***weight*** *of e. The* ***weight*** *of a walk in G is the sum of the weights of the edges traversed by the walk, counting repetitions.*

Formally a weighted graph G may be characterized as an ordered quadruple $G = (V, E, \iota, \omega)$, where as usual ι is the graph's incidence function, and $\omega : E \to \mathbb{R}$ is an **edge-weight function** that for each edge $e \in E$ returns an associated weight $\omega(e)$ that is some real number.[9] The weight of an edge or path is also called the **length** of the edge or path, and thus it is natural to define the **distance** between vertices $u, v \in V$, denoted by $d^\omega(u, v)$, to be the length of the shortest u, v-path in G. If no u, v-path exists in G, then the distance is "infinite" and we write $d^\omega(u, v) = \infty$. Thus $d^\omega(u, v)$ will be real-valued for all $u, v \in V$ in a connected weighted graph. The length (or weight) of a trivial path is defined to be 0, as in an unweighted graph. Finally, if W is a walk with vertex-edge sequence $v_0 e_1 v_1 \cdots e_n v_n$, then $\omega(W)$ denotes the weight of W, so that

$$\omega(W) = \omega(v_0 e_1 v_1 \cdots e_n v_n) = \sum_{k=1}^{n} \omega(e_k).$$

Example 10.100. In Figure 18 is a drawing of a weighted simple graph G, with the weight of each edge displayed at approximately the midpoint of the edge. The weight of edge ad is 2, for instance. If ω is the edge-weight function for G, then $\omega(ad) = 2$, $\omega(de) = 1$, $\omega(ez) = 13$, and so on. The weight of the path with vertex sequence $adez$ is

$$\omega(ad) + \omega(de) + \omega(ez) = 2 + 1 + 13 = 16.$$

[9] We use the Greek letter ω here for its resemblance to the letter w, for "weight," thereby leaving w free to denote a vertex.

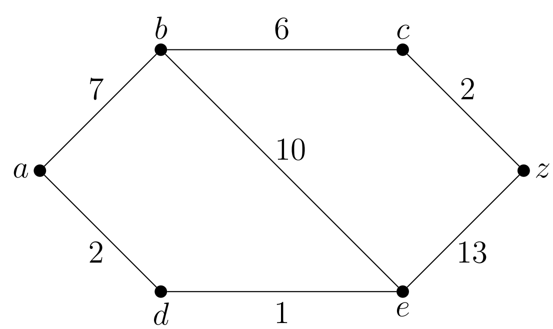


Figure 18.

This is not the shortest a, z-path in G, however. The path $abcz$ has length $7 + 6 + 2 = 15$, which is seen to be the shortest path by inspection, and thus the distance between a and z is $d^\omega(a, z) = 15$. ■

In Example 10.100 we found the length of the shortest path from a to z in the graph depicted in Figure 18 by inspection (or what might be called "brute force"). In the next example we shall approach the problem of finding the distance between a and z methodically, using an approach known as Dijkstra's algorithm. The treatment will be heuristic to start, favoring verbal descriptions over the use of symbols or code.

Example 10.101. Let G be the graph in Figure 18. To find the distance (i.e. the length of the shortest path) between vertices a and z, we apply a method that we might use if we knew the lengths of only those edges that are incident to vertices that we have visited via some path. The first vertex we visit is a.

We first find the closest vertex to a. This is done by examining all paths that can be constructed by the following procedure: For $x \in \{a\}$ find the shortest a, x-path, then concatenate this a, x-path with an edge uv for which $u \in \{a\}$ and $v \notin \{a\}$. Letting $x = a$ (our only choice), the shortest a, a-path is the trivial path with vertex sequence a, so that $u = a$ since edge uv must be incident to a, and then v may be either b or d. If $v = b$, the path ab of length 7 results; and if $v = d$, the path ad of length 2 results. Therefore d is the closest vertex to a, and the shortest a, d-path is ad with $d^\omega(a, d) = 2$.

We next find the second closest vertex to a. This is done by examining all paths that can be constructed as follows: For $x \in \{a, d\}$ find the shortest a, x-path, then concatenate this a, x-path with an edge uv for which $u \in \{a, d\}$ and $v \notin \{a, d\}$. Letting $x = a$ yields the path ab of length 7. Letting $x = d$, the shortest a, d-path is ad, and then $uv = de$ so that the path ade of length 3 results. Therefore e is the second closest vertex to a, and the shortest a, e-path is ade with $d^\omega(a, e) = 3$.

Now to find the third closest vertex to a. We do this by examining all paths that can be constructed as follows: For $x \in \{a, d, e\}$ find the shortest a, x-path, then concatenate this a, x-path with an edge uv for which $u \in \{a, d, e\}$ and $v \notin \{a, d, e\}$. Letting $x = a$ yields path ab (length 7); letting $x = d$ yields no path; and letting $x = e$ yields paths $adeb$ (length 13) and $adez$ (length 16). Therefore b is the third closest vertex to a, and the shortest a, b-path is ab with $d^\omega(a, b) = 7$.

Next we find the fourth closest vertex to a. We do this by examining all paths that can be constructed as follows: For $x \in \{a, d, e, b\}$ find the shortest a, x-path, then concatenate this a, x-path with an edge uv for which $u \in \{a, d, e, b\}$ and $v \notin \{a, d, e, b\}$. Letting $x = a$ or $x = d$ yields no path; letting $x = e$ yields path $adez$ (length 16); letting $x = b$ yields path abc (length 13). Therefore c is the fourth closest vertex to a, and the shortest a, c-path is abc with $d^\omega(a, c) = 13$.

Finally we find the fifth closest vertex to a. We do this by examining all paths that can be constructed as follows: For $x \in \{a, d, e, b, c\}$ find the shortest a, x-path, then concatenate this a, x-path with an edge uv for which $u \in \{a, d, e, b, c\}$ and $v \notin \{a, d, e, b, c\}$ (so $v = z$). Letting x be a, d, or b yields no path; letting $x = e$ yields path $adez$ (length 16); letting $x = c$ yields path $abcz$ (length 15). Therefore z is the fifth closest vertex to a (as we could have guessed by a process of elimination), and the shortest a, z-path is $abcz$ with $d^\omega(a, z) = 15$. ■

Algorithm Dijkstra Dijkstra's Algorithm

Input: G [weighted connected simple graph with $\omega > 0$ and vertices $a = v_0, v_1, \ldots, v_n = z$]

for $i := 1$ **to** n **do**
  $\ell(v_i) := \infty$
end for
$\ell(a) := 0$
$S := \varnothing$
while $z \notin S$ **do** ▷ $V(G) - S \neq \varnothing$ instead of $z \notin S$ makes all $v \neq a$ targets.
  $u := v \notin S$ with $\ell(v)$ is minimal
  $S := S \cup \{u\}$
  for all $v \notin S$ **do**
    if $\ell(u) + \omega(uv) < \ell(v)$ **then**
      $\ell(v) := \ell(u) + \omega(uv)$
    end if
  end for
end while

Output: $\ell(z)$

When applying Dijkstra's algorithm to find the length of the shortest path between vertices a and z in a weighted graph G, we call a the **source** vertex and z the **target** vertex. It's noteworthy that the algorithm not only finds the distance from a source to a target, but also the distances between the source and every vertex that is closer to the source than the target. In Example 10.101, distances $d^\omega(av)$ were found for *all* $v \in V(G)$ because the target z happened to be farther from a than all other vertices in G. In general one could execute the algorithm without a specific target in mind, specifying only a source a, until the distances from a to all over vertices in a weighted graph are found. One thing Dijkstra's algorithm does require, however, is that all edge weights be *positive* real numbers. Given any weighted graph G with edge-weight function ω, we write $\omega > 0$ to indicate that $\omega : E(G) \to (0, \infty)$; that is, $\omega(e) > 0$ for all $e \in E(G)$.

The pseudocode for Dijkstra's algorithm is given by Algorithm Dijkstra, with source vertex being a and target vertex being z. It begins by assigning a **label** $\ell(v)$ to every vertex $v \in V(G)$ that is also called the **tentative distance** between a and v. Before the first iteration of the *while* loop the label $\ell(v)$ is the length of the shortest path from a to v consisting only of vertices in the set S together with v itself. However, S is empty to start, so no such a, v-path exists unless $v = a$, and hence $\ell(v) = \infty$ if $v \neq a$, and $\ell(a) = 0$. Once this is done, the *while* loop is executed.

Let S_k be the set S in the algorithm during the kth iteration of the *while* loop, and let $\ell_k(v)$ be the label assigned to v during the kth iteration of the *while* loop. Then $S_0 = \varnothing$, and since $\ell_0(a) = 0$ is minimal, the first iteration of the *while* loop starts by designating $S_1 = S_0 \cup \{a\} = \{a\}$, followed by a *for* loop assigns new labels to all vertices except a. Since $z \neq a$, the *while* loop then repeats, adding a vertex u with minimal label $\ell_1(u)$ to S_1 to form $S_2 = \{a, u\}$, after which the *for* loop again assigns updated labels.

After the $(k-1)$st iteration of the *while* loop, there will be a vertex $u \notin S_{k-1}$ with $\ell_{k-1}(u)$ minimal, and so the kth iteration will commence by defining $S_k = S_{k-1} \cup \{u\}$. (If two or more vertices have the same minimal label value, the algorithm will choose only one of the vertices to be in S_k, and leave the others out.) The *for* loop nested within the *while* loop is then executed, comparing $\ell(u)+\omega(uv)$ with $\ell(v)$ to determine whether $\ell(u)+\omega(uv)<\ell(v)$ is true or false. If $uv\notin E(G)$, then $\omega(uv)$ is undefined and so $\ell(u)+\omega(uv)<\ell(v)$ is judged to be false. In keeping with our use of the symbol ∞ thus far, we will set $\omega(uv)=\infty$ if u and v are not adjacent in G. The operating rules for ∞ are: $\infty+x=x+\infty=\infty$ for all $x\in\mathbb{R}\cup\{\infty\}$, and $x<\infty$ for all $x\in\mathbb{R}$. In particular we consider $\infty<\infty$ to be false.

Suppose $v\notin S_k$. Then, in the course of the kth iteration of the *while* loop, the *for* loop will update the label assigned to v. The *if-then* statement in the *for* loop ensures that

$$\ell_k(v)=\min\{\ell_{k-1}(v),\ell_{k-1}(u)+\omega(uv)\},$$

where again u is the vertex for which $\ell_{k-1}(u)$ is minimal among all labels $\ell_{k-1}(w)$ for $w\notin S_{k-1}$. We will see in the proof of Theorem 10.102 that $\ell_k(v)$ is in fact the length of the shortest path from a to v containing only vertices in S_k (except for v).

If $V(G)=\{v_0,v_1,\dots,v_n\}$ with $a=v_0$ and $z=v_n$, as in Algorithm Dijkstra, then the *while* loop must repeat at most $n+1$ times before the algorithm terminates and returns the length of the shortest possible a,z-path, since after the $(n+1)$st iteration all $n+1$ vertices of the graph will be in the set S. Of course, it's more likely that z will be added to the set S earlier than this, in which case the algorithm will terminate when some vertices are still not in S.

To have the algorithm find the length of the shortest a,v-path for *all* $v\in V(G)$, one need only replace the condition $z\notin S$ in the first line of the *while* block with something like $S\neq V(G)$ or $S-V(G)\neq\varnothing$, and then have the algorithm output the final labels of all vertices. This effectively makes all vertices of the graph targets, and not just the vertex z.

Theorem 10.102. *Dijkstra's algorithm finds the length of the shortest path between a source vertex a and target vertex z in a connected simple undirected weighted graph.*

Proof. For $k\geq 0$ let ℓ_k and S_k be defined as before. Define the predicates

$P_1(k)$: "For all $v\in S_k$, $\ell_k(v)$ is the length of the shortest a,v-path."

$P_2(k)$: "For all $v\notin S_k$, $\ell_k(v)$ is the length of the shortest a,v-path for which all vertices but v are in S_k."

Also let $P(k)$ be the predicate "After the kth iteration of the *while* loop in Dijkstra's algorithm, $P_1(k)$ and $P_2(k)$ are true." To prove the theorem, we first prove by induction that $P(k)$ is true for all $k\geq 0$.[10]

When $k=0$, so that the *while* loop has not yet been executed even once, we have $S_0=\varnothing$, and so $P_1(0)$ is vacuously true. The initial labels $\ell_0(a)=0$ and $\ell_0(v)=\infty$ for $v\neq a$, meanwhile, show $P_2(0)$ to also be true. Hence $P(0)$ is true, and therefore the basis case is true.

For the inductive step, suppose $P(k)$ is true for some $k\geq 0$. We will first show that $P_1(k+1)$ is true. By $P_1(k)$, each $v\in S_k$ has label $\ell_k(v)$ which is the length of the shortest possible a,v-path, and since the algorithm does not update the label of any $v\in S_k$, it is established that every $v\in S_{k+1}$ likewise has a label that equals the length of the shortest a,v-path *with*

[10] Certainly the theorem could proved if just $P_1(k)$ alone is shown to be true for all $k\geq 0$, but this is more easily accomplished by incorporating $P_2(k)$ into the inductive hypothesis.

the exception of the vertex u with minimal label $\ell_k(u)$ that is added to the set S to form the set S_{k+1}. Since $u \in S_{k+1}$, the label $\ell_k(u)$ is not updated during the $(k+1)$st iteration, and so it remains to show that $\ell_k(u)$ is the length of the shortest a,u-path. By way of contradiction suppose it is not, so that there exists an a,u-path P_{au} of length less than $\ell_k(u)$. Since $u \notin S_k$, $P_2(k)$ implies that $\ell_k(u)$ is the length of the shortest a,u-path having all vertices but u in S_k, so that P_{au} must contain at least one vertex $w \neq u$ such that $w \notin S_k$, and we will assume w is the first such vertex in the vertex sequence for P_{au}. The a,w-path P_{aw} that is a subpath of P_{au} is certainly shorter than P_{au}, and except for w, P_{aw} contains only vertices in S_k. Since by $P_2(k)$ the label $\ell_k(w)$ is the length of the shortest a,w-path having all vertices but w in S_k, it follows that $\ell_k(w)$ is less than or equal to the length of P_{aw}, and hence $\ell_k(w) < \ell_k(u)$. But this contradicts the hypothesis that $\ell_k(u)$ is minimal. Therefore $\ell_k(u) = d^\omega(au)$, and $P_1(k+1)$ is true.

We now prove $P_2(k+1)$. Let $v \notin S_{k+1}$, and let u be as before (that is, $u \notin S_k$ and $u \in S_{k+1}$). Denote by P_{av} the shortest a,v-path with all vertices but v in S_{k+1}. To show is that the length of P_{av} equals $\ell_{k+1}(v)$. We first consider the case in which u is not a vertex in P_{av}. Then P_{av} is a shortest a,v-path with all vertices but v in S_k. On the other hand, $P_2(k)$ implies that $\ell_k(v)$ is the length of the shortest a,v-path with all vertices but v in S_k, and so the length of P_{av} equals $\ell_k(v)$. Now, suppose $\ell_{k+1}(v) < \ell_k(v)$. This indicates that the algorithm updated $\ell(v)$ during the $(k+1)$st iteration, with $\ell_{k+1}(v) := \ell_k(u) + \omega(uv)$, and so $\ell_k(u) + \omega(uv) < \ell_k(v)$. Now, by $P_2(k)$, $\ell_k(u)$ is the length of a shortest a,u-path with vertex sequence P_{au} having all vertices but u in S_k, and thus $P_{au} * uv$ is an a,v-path having all vertices but v in S_{k+1}. The length of $P_{au} * uv$ is $\ell_k(u) + \omega(uv)$, which is less than the length of P_{av}. As this is impossible, and $\ell_{k+1}(v) > \ell_k(v)$ is also impossible (labels never increase in value), we conclude that $\ell_{k+1}(v) = \ell_k(v)$, and therefore $\ell_{k+1}(v)$ equals the length of P_{av}.

We finally turn to the case wherein the shortest path from a to v with all vertices but v in S_{k+1} must pass through u. Again $\ell_k(v)$ is the length of the shortest a,v-path with all vertices but v in S_k, which we denote by P'_{av}. Since $u \notin S_k$, the path P'_{av} does not pass through u, and so P'_{av} must be longer than the concatenation of the shortest a,u-path with all vertices but u in S_k with the path uv, which has length $\ell_k(u) + \omega(uv)$. Thus $\ell_k(u) + \omega(uv) < \ell_k(v)$, and so Dijkstra's algorithm defines $\ell_{k+1}(v) = \ell_k(u) + \omega(uv)$. Since $\ell_k(u)$ is the length of the shortest a,u-path with all vertices but u in S_k, it follows that $\ell_k(u) + \omega(uv)$ is the length of the shortest a,v-path with all vertices but v in S_{k+1}. We conclude once more that $\ell_{k+1}(v)$ equals the length of P_{av}, showing that $P_2(k+1)$ is true, and therefore $P(k+1)$ is true.

Finally, Algorithm Dijkstra terminates after the completion of some iteration of the *while* loop in which z is included in S and a final updating of labels is done, returning as output the label $\ell(z)$. Assuming this occurs during the kth iteration, it follows by $P_1(k)$ that the output $\ell(z)$ is the length of the shortest a,z-path in G. ■

Theorem 10.103. *For a graph with n vertices, the time complexity of Dijkstra's algorithm is at most $O(n^2)$.*

Proof. Suppose a graph G has n vertices. At worst the target vertex z is added to S last, and since each iteration of the *while* loop adds one vertex to S, starting with the empty set, the *while* loop will need to be iterated no more than n times. Now, for each iteration, at worst

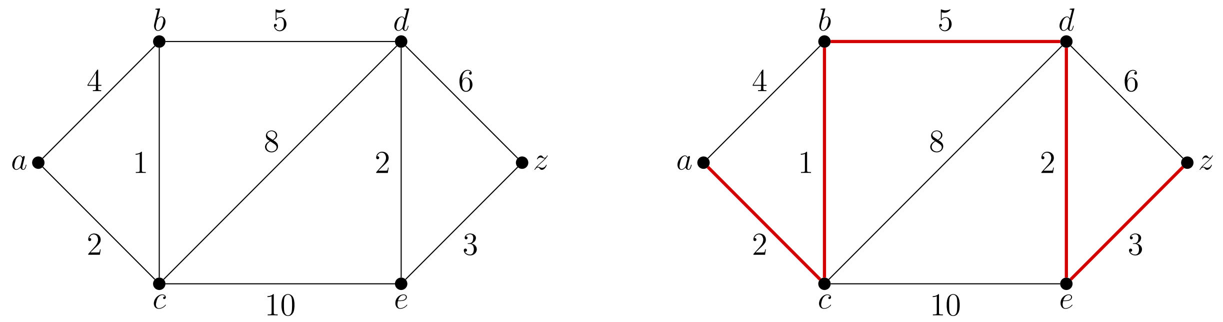


FIGURE 19. At left: the weighted graph G; at right: the shortest a, z-path in G.

$n-1$ comparisons must be done to find a vertex with a minimal label,[11] whereafter the *for* loop will at worst perform $n-1$ additions and $n-1$ comparisons. Thus each of the at most n iterations of the *while* loop will perform at most $3(n-1)$ operations, for a total of at most $3n(n-1)$ operations. Finally, let $T(n)$ denote the time necessary for the algorithm to complete its task when given a graph with n vertices. Assuming each operation (whether an addition or a comparison) takes time c to execute, we have $T(n) \leq 3cn(n-1) < 3cn^2$ for all $n \geq 2$, and therefore $T(n)$ is at most $O(n^2)$. ■

Example 10.104. Here we shall use Dijkstra's algorithm, as presented by the pseudocode of Algorithm Dijkstra, to find the length of the shortest a, z-path in the weighted graph G with edge-weight function ω and $V(G) = \{a, b, c, d, e, z\}$ depicted in Figure 19. The procedure begins by declaring $\ell_0(a) := 0$, $\ell_0(v) := \infty$ for all $v \neq a$, and $S_0 := \varnothing$.

1st iteration of the while loop: We have $a \notin S_0$ with $\ell_0(a)$ minimal, so $S_1 := S_0 \cup \{a\} = \{a\}$, whereafter the *for* loop is executed:

$$\ell_0(a) + \omega(ab) = 4 < \infty = \ell_0(b) \quad \hookrightarrow \quad \ell_1(b) := 4.$$

$$\ell_0(a) + \omega(ac) = 2 < \infty = \ell_0(c) \quad \hookrightarrow \quad \ell_1(c) := 2.$$

$$\ell_0(a) + \omega(av) = \infty \not< \infty = \ell_0(v) \quad \hookrightarrow \quad \ell_1(v) := \infty \text{ for } v = d, e, z.$$

2nd iteration: $\ell_1(c) = 2$ is minimal, so $S_2 = S_1 \cup \{c\} = \{a, c\}$. For those vertices not in S_2 we have:

$$\ell_1(c) + \omega(cb) = 3 < 4 = \ell_1(b) \quad \hookrightarrow \quad \ell_2(b) := 3.$$

$$\ell_1(c) + \omega(cd) = 10 < \infty = \ell_1(d) \quad \hookrightarrow \quad \ell_2(d) := 10.$$

$$\ell_1(c) + \omega(ce) = 12 < \infty = \ell_1(e) \quad \hookrightarrow \quad \ell_2(e) := 12.$$

$$\ell_1(c) + \omega(cz) = \infty \not< \infty = \ell_1(z) \quad \hookrightarrow \quad \ell_2(z) := \infty.$$

3rd iteration: $\ell_2(b) = 3$ is minimal, so $S_3 = S_2 \cup \{b\} = \{a, b, c\}$. For those vertices not in S_3 we have:

$$\ell_2(b) + \omega(bd) = 8 < 10 = \ell_2(d) \quad \hookrightarrow \quad \ell_3(d) := 8.$$

[11] This assumes vertex labels are examined one by one, though there are sorting algorithms that can do the task more efficiently.

k	$\ell_k(a)$	$\ell_k(b)$	$\ell_k(c)$	$\ell_k(d)$	$\ell_k(e)$	$\ell_k(z)$
0	0	∞	∞	∞	∞	∞
1	0	4	2	∞	∞	∞
2	0	3	2	10	12	∞
3	0	3	2	8	12	∞
4	0	3	2	8	10	14
5	0	3	2	8	10	13

FIGURE 20. Vertex labels after the kth iteration of the *while* loop.

$$\ell_2(b) + \omega(be) = \infty \not< 12 = \ell_2(e) \quad \hookrightarrow \quad \ell_3(e) := 12.$$

$$\ell_2(b) + \omega(bz) = \infty \not< \infty = \ell_2(z) \quad \hookrightarrow \quad \ell_3(z) := \infty.$$

4th iteration: $\ell_3(d) = 8$ is minimal, so $S_4 = S_3 \cup \{d\} = \{a, b, c, d\}$. For those vertices not in S_4 we have:

$$\ell_3(d) + \omega(de) = 10 < 12 = \ell_3(e) \quad \hookrightarrow \quad \ell_4(e) := 10.$$

$$\ell_3(d) + \omega(dz) = 14 < \infty = \ell_3(z) \quad \hookrightarrow \quad \ell_4(z) := 14.$$

5th iteration: $\ell_4(e) = 10$ is minimal, so $S_5 = S_4 \cup \{e\} = \{a, b, c, d, e\}$. For $z \notin S_4$ we have:

$$\ell_4(e) + \omega(ez) = 10 + 3 = 13 < 14 = \ell_4(z) \quad \hookrightarrow \quad \ell_5(z) := 13.$$

As with Example 10.101, it so happens that the target z is the last vertex to be included in the set S, and so the algorithm in fact finds the length of the shortest a, v-path, which is the distance $d^\omega(a, v)$, for *all* $v \in V(G)$. The results of each iteration are summarized in the table in Figure 20. The bottom row of the table gives $\ell_5(v)$ for each vertex v in the graph, where $\ell_5(v) = d^\omega(a, v)$. ■

It has likely not escaped the notice of the reader that though Algorithm Dijkstra returns as output the length of a shortest a, z-path, it does not specify any such path. While Dijkstra is quite close in structure to the original algorithm conceived by Edsger Dijkstra in 1956, many "modified" forms of the procedure have since been devised that do, in fact, furnish a compass to trace shortest paths along their vertex sequences. Such a compass is sometimes called a **pointer**, and we present Algorithm Dijkstra2 as an example of a shortest-path algorithm that returns pointers as well as distances as outputs.

In addition to returning pointers, Dijkstra2 has a few other modifications relative to Dijkstra that are made for purposes of demonstating the variety of ways that a computer program dedicated to performing a particular task may be structured. First, the initialization phase of Dijkstra2 places the source vertex v_1 in the set S right away, before the first iteration of the *while* loop. Second, the condition $z \notin S$ of the *while* loop in Dijkstra has been replaced by $V(G) - S \neq \varnothing$, which declares all vertices $v_i \neq v_1$ to be targets, and not merely some particular vertex z. Third and most importantly, there is the pointer output $p(v_i)$, which will be an ordered pair of the form $(d^\omega(v_1, v_i), u)$; the first component in the pair will be the length of a shortest v_1, v_i-path, and the second component will be the vertex u that *precedes* v_i in the

Algorithm Dijkstra2 Dijkstra's Algorithm with Pointers

Input: G [weighted connected simple graph with $\omega > 0$ and vertices $v_1, v_2, \ldots, v_n$]

1: $\ell(v_1) := 0$, $S := \{v_1\}$, $u := v_1$
2: **for** $i := 2$ **to** n **do**
3: $\quad \ell(v_i) := \infty$
4: **end for**
5: **while** $V(G) - S \neq \varnothing$ **do**
6: $\quad$ **for** $i := 2$ **to** n **do**
7: $\quad\quad$ **while** $v_i \in V(G) - S$ and $uv_i \in E(G)$ **do**
8: $\quad\quad\quad$ **if** $\ell(u) + \omega(uv_i) < \ell(v_i)$ **then**
9: $\quad\quad\quad\quad \ell(v_i) := \ell(u) + \omega(uv_i)$ and $p(v_i) := (\ell(v_i), u)$
10: $\quad\quad\quad$ **end if**
11: $\quad\quad$ **end while**
12: $\quad$ **end for**
13: $\quad m := \min\{\ell(v_i) : v_i \in V(G) - S\}$
14: $\quad$ **for** $i := 2$ **to** n **do**
15: $\quad\quad$ **while** $v_i \in V(G) - S$ **do**
16: $\quad\quad\quad$ **if** $\ell(v_i) = m$ **then**
17: $\quad\quad\quad\quad S := S \cup \{v_i\}$, $u := v_i$, $m := m - 1$
18: $\quad\quad\quad$ **end if**
19: $\quad\quad$ **end while**
20: $\quad$ **end for**
21: **end while**
22: **for** $i := 2$ **to** n **do**
23: $\quad$ **Output** $p(v_i)$
24: **end for**

vertex sequence of a shortest v_1, v_i-path (so that u is the penultimate vertex in some sequence $v_1 \cdots v_i$).

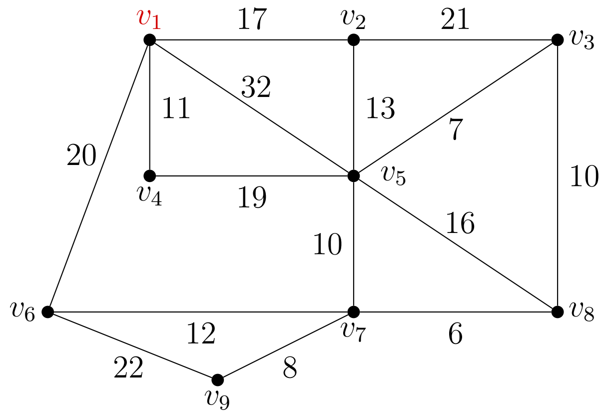
Example 10.105. The weighted graph G at left in Figure 21 has $V(G) = \{v_1, v_2, \ldots, v_9\}$ and $\omega > 0$. We use Algorithm Dijkstra2 to find, for each $2 \leq i \leq 9$, the length of the shortest possible v_1, v_i-path, and also a vertex u_i that immediately precedes v_i along such a path. The mechanics of the routine are largely the same as in Example 10.104, but there are some important differences. Here we shall run through the first iteration of the *while* loop that begins on line 5 of Dijkstra2, with the results for all eight iterations summarized in the table in Figure 22.

For the first iteration of the *while* loop, we find that the *while* loop beginning on line 7 is executed only for $i \in \{2, 4, 5, 6\}$, since otherwise $uv_i = v_1 v_i \in E(G)$ is false. Since $u = v_1$, we have:

$$\ell_0(v_1) + \omega(v_1 v_2) = 17 < \infty = \ell_0(v_2) \quad \hookrightarrow \quad \ell_1(v_2) := 17,\ p(v_2) := (17, v_1),$$

$$\ell_0(v_1) + \omega(v_1 v_4) = 11 < \infty = \ell_0(v_4) \quad \hookrightarrow \quad \ell_1(v_4) := 11,\ p(v_4) := (11, v_1),$$

$$\ell_0(v_1) + \omega(v_1 v_5) = 32 < \infty = \ell_0(v_5) \quad \hookrightarrow \quad \ell_1(v_5) := 32,\ p(v_5) := (32, v_1),$$



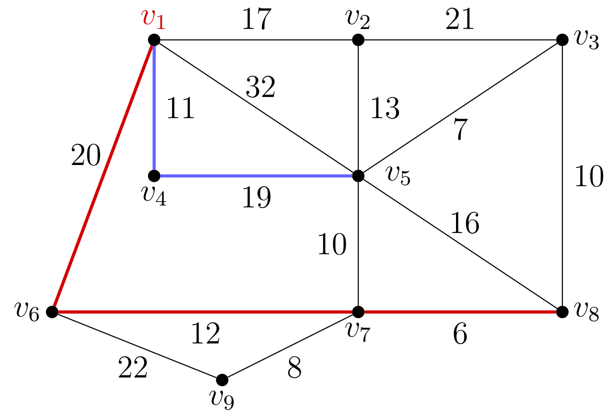


FIGURE 21. At left: the weighted graph G; at right: the shortest v_1, v_5-path in G prescribed by Dijkstra2, and the shortest v_1, v_8-path.

$$\ell_0(v_1) + \omega(v_1 v_6) = 20 < \infty = \ell_0(v_6) \quad \hookrightarrow \quad \ell_1(v_6) := 20,\ p(v_6) := (20, v_1),$$

and $\ell_1(v_i) = \ell_0(v_i) = \infty$ for $i \in \{3, 7, 8, 9\}$. Thus $m := \ell_1(v_4) = 11$, so that $S_1 := S_0 \cup \{v_4\} = \{v_1, v_4\}$, $u := v_4$, and $m := 11 - 1 = 10$. (This decrease in the value of m ensures that if there is more than one vertex in the set $\{\ell(v_i) : v_i \in V(G) - S\}$ with label equal to m at line 13, then it is only the vertex with the *smallest index* that is included in the set S at line 17.) The *while* loop now repeats.

The bottom row of the table in Figure 22 is the only output the algorithm returns, in accordance with the *for* loop on line 22 of Dijkstra2. Along this row, for each $v_i \in V(G) - \{v_1\}$, is a pair of data: the first datum is $d^\omega(v_1, v_i)$, and the second datum is the vertex u that immediately precedes v_i on a v_1, v_i-path of length $d^\omega(v_1, v_i)$. For instance we have $p(v_8) = (38, v_7)$, which indicates that the shortest v_1, v_8-path has length 38, and v_7 immediately precedes v_8 along such a path. To find what precedes v_7 on this path, we note from the table that $p(v_7) = (32, v_6)$, and so the shortest v_1, v_7-path, which must necessarily be a subpath of the shortest v_1, v_8-path, has v_6 immediately preceding v_7. What comes before v_6 is v_1, as indicated by $p(v_6) = (20, v_1)$. Therefore $v_1 v_6 v_7 v_8$ is the vertex sequence of a shortest v_1, v_8-path. We find in the same way that $v_1 v_4 v_5$ is a shortest v_1, v_5-path, though it is noteworthy that $v_1 v_2 v_5$ is another v_1, v_5-path having the same length. The algorithm chose $v_1 v_4 v_5$ because $d^\omega(v_1, v_4) < d^\omega(v_1, v_2)$. This is a hallmark of a so-called "greedy algorithm." ∎

We conclude this section with a whirlwind tour of the so-called **traveling salesman problem** (abbreviated **TSP**). There is a salesman who must make a round trip that visits n cities, including the city from which the salesman departs. For any two such cities that are joined by a travel route, the length of the route is known. The problem asks: If the salesman is to visit each of the n cities exactly once, then what is the minimum possible distance of such a round trip?

One could just as easily substitute "cost" for "distance," and ask what the minimum cost of such a round trip will be. This would require knowledge of the cost of traveling between any two cities for which there exists a means of travel. Or perhaps time is of the essence, and the

k	$p(v_2)$	$p(v_3)$	$p(v_4)$	$p(v_5)$	$p(v_6)$	$p(v_7)$	$p(v_8)$	$p(v_9)$
0	∞	∞	∞	∞	∞	∞	∞	∞
1	$(17, v_1)$	∞	$(11, v_1)$	$(32, v_1)$	$(20, v_1)$	∞	∞	∞
2	$(17, v_1)$	∞	$(11, v_1)$	$(30, v_4)$	$(20, v_1)$	∞	∞	∞
3	$(17, v_1)$	$(38, v_2)$	$(11, v_1)$	$(30, v_4)$	$(20, v_1)$	∞	∞	∞
4	$(17, v_1)$	$(38, v_2)$	$(11, v_1)$	$(30, v_4)$	$(20, v_1)$	$(32, v_6)$	∞	$(42, v_6)$
5	$(17, v_1)$	$(37, v_5)$	$(11, v_1)$	$(30, v_4)$	$(20, v_1)$	$(32, v_6)$	$(46, v_5)$	$(42, v_6)$
6	$(17, v_1)$	$(37, v_5)$	$(11, v_1)$	$(30, v_4)$	$(20, v_1)$	$(32, v_6)$	$(38, v_7)$	$(40, v_7)$
7	$(17, v_1)$	$(37, v_5)$	$(11, v_1)$	$(30, v_4)$	$(20, v_1)$	$(32, v_6)$	$(38, v_7)$	$(40, v_7)$
8	$(17, v_1)$	$(37, v_5)$	$(11, v_1)$	$(30, v_4)$	$(20, v_1)$	$(32, v_6)$	$(38, v_7)$	$(40, v_7)$

FIGURE 22. Values of $p(v_i) = (\ell(v_i), u)$ for iteration $0 \leq k \leq 8$ of the outer *while* loop of Dijkstra2. We set $p(v_i) = \infty$ if $\ell(v_i) = \infty$, and leave $p(v_1)$ undefined.

salesman wishes to minimize the time traveled during the round trip. In any case, it is assumed it is possible to visit every city exactly once in the course of the round trip, which is to say the graph G whose vertices represent the cities and whose edges represent travel routes between cities must possess a Hamiltonian cycle. Indeed, G will be a weighted graph, with the weight of each edge representing distance, cost, time, or some other quantity one might wish to minimize. To solve the TSP, the most straightforward way is to identify all the possible Hamiltonian cycles in G, calculate the weight of each, and identify the cycle with the minimum weight. We call this method of solution the **naive approach**.

The naive approach to the TSP is simple enough to understand in principle, but as an algorithm its time complexity is terrible. Suppose a simple undirected weighted graph G with n vertices is complete, so that $G \simeq K_n$. Since K_n is Hamiltonian by Proposition 10.98, so too is G, and thus we're assured there exists at least one Hamiltonian cycle in G. Let c_1 be the starting vertex (and hence ending vertex) of a Hamiltonian cycle in G. Being complete, every vertex in G is a neighbor of every other vertex, and so, starting at c_1, we commence to count the total number of Hamiltonian cycles in the graph. At c_1 there are $n-1$ choices for the second vertex c_2 in the cycle. At c_2 there are $n-2$ remaining choices for the third vertex c_3, and at c_3 there are $n-3$ choices for the fourth vertex c_4, and so on. Since there are n choices for the first vertex c_1, by the multiplication rule of counting there are $n!$ Hamiltonian cycles with distinct vertex sequences.

However, for purposes of the TSP it is immaterial what the starting vertex of the cycle is chosen to be: If the cycle C with vertex sequence $c_1c_2\cdots c_nc_1$ has weight w, then the cycles $c_2c_3\cdots c_nc_1c_2$, $c_3c_4\cdots c_nc_1c_2c_3$, and so on to $c_nc_1c_2\cdots c_n$ will all have weight w. This means there is no need to designate a "home base" for the saleman's round trip, since whatever the minimum possible distance for a round trip commencing at one city is, it will be the same for a round trip commencing at any other city on the itinerary. This leaves us with $(n-1)!$ possible Hamiltonian cycles to consider, or what one might call equivalence classes of Hamiltonian cycles, with the equivalence class for $c_1c_2\cdots c_nc_1$ defined to be

$$[c_1c_2\cdots c_nc_1] = \{c_ic_{i+1}\cdots c_nc_1c_2\cdots c_i : 1 \leq i \leq n\}.$$

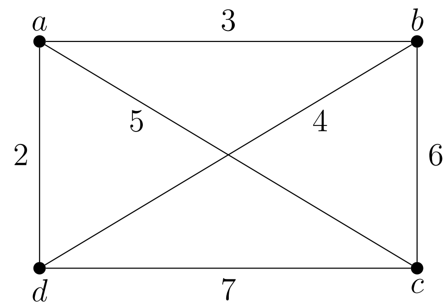


FIGURE 23.

(This is to say that two cycles are equivalent if and only if the vertex sequence of one is a circular permutation of the vertex sequence of the other.) As a further economy, we note that once a round trip of minimal distance is found, the round trip along the same route but in the opposite direction will have the same minimal distance. That is, cycles $c_1c_2\cdots c_nc_1$ and $c_1c_n\cdots c_2c_1$ must always have the same weight, and therefore the total number of Hamiltonian cycles to consider is cut from $(n-1)!$ to $(n-1)!/2$. Nevertheless, the numbers involved quickly become enormous even for modest values of n. If $n = 50$, then $(n-1)!/2 \approx 3.04 \times 10^{62}$. In this case, if we calculated the weights of a quadrillion cycles per second since the time of the Big Bang (currently thought to be about 13 billion years ago), then the fraction of the task that would be completed by the present day would be less than 10^{-29}.

For a graph G with $|V(G)| = n$, we see that the naive approach to the TSP has time complexity of order $O(n!)$, and there is no known algorithm that can solve the problem more efficiently. To find an algorithm with polynomial time complexity (i.e. time complexity of order $O(n^k)$ for some positive integer k) is a long-sought holy grail of theoretical computer science, but it is not even known if such an algorithm exists. There are, however, algorithms that can find approximate solutions to the TSP fairly fast; that is, they find Hamiltonian cycles having weights "close" to the minimum.

Example 10.106. Solve the traveling salesman problem for the graph G in Figure 23.

Solution. We can let vertex a be the starting vertex of every Hamiltonian cycle. The full list of such cycles is:

$$abcda, \quad abdca, \quad acbda, \quad acdba, \quad adbca, \quad adcba.$$

However, half of these cycles pass through the vertices of G in the reverse order of the other half: $abcda$ and $adcba$, $abdca$ and $acdba$, and $acbda$ and $adbca$. Since reversing the direction of travel along any given walk results in a new walk with the same weight as the old, we need only find the weights of $abcda$, $abdca$, and $acbda$. We now find the weights of these three cycles:

$$\omega(abcda) = 3 + 6 + 7 + 2 = 18,$$

$$\omega(abdca) = 3 + 4 + 7 + 5 = 19,$$

$$\omega(acbda) = 5 + 6 + 4 + 2 = 17.$$

We conclude that the the minimum weight of a Hamiltonian cycle in G is 17, and $acbda$ is one such cycle. ∎

The algorithms presented in this section are easily adapted to work for weighted simple digraphs, but this will not be discussed here.

11

Trees

11.1 – Properties of Trees

To begin our study of trees, recall that a graph having no subgraph that is a nontrivial cycle is called acyclic. An acyclic graph cannot have loops (cycles of length 1) or parallel edges (cycles of length 2), and so is necessarily a simple graph.

Definition 11.1. *A* ***tree*** *is a connected acyclic graph. A* ***forest*** *is an acyclic graph, connected or not. A vertex of degree 1 in a tree or forest is a* ***leaf****.*

The contrapositive of Lemma 10.92 informs us that if a graph G is acyclic, then there exists some $v \in V(G)$ such that $d(v) < 2$. If G is connected as well, so that it is a tree, then it must be that $d(v) = 1$ and hence vertex v is a leaf. In fact we can be sure that any tree having an edge must possess at least *two* leaves.

Proposition 11.2. *Any tree with at least one edge has at least two leaves.*

Proof. Let T be a tree with at least one edge, and let $P = v_1v_2\cdots v_n$ be a path of maximum length in T. We note that the length of P must be at least 1 since T has an edge, and thus $v_n \neq v_1$. Now, suppose v_1 is not a leaf, so that $d(v_1) \geq 2$, and there is some vertex $u \neq v_2$ that is adjacent to v_1. There are two possibilities: $u \notin V(P)$ or $u \in V(P)$. If $u \notin V(P)$, then $P' = uv_1v_2\cdots v_n$ is a path in T of greater length than P—a contradiction. If $u \in V(P)$, then $u = v_k$ for some $2 \leq k \leq n$, and thus $v_1v_2\cdots v_kv_1$ is a cycle in T—again a contradiction. We conclude that v_1 must be a leaf, and by the same argument so too must v_n. Therefore T has two leaves. ∎

Proposition 11.3. *If T is a tree with $|V(T)| = n$, then $|E(T)| = n - 1$.*

Proof. The proof will be by induction. Certainly if T is a tree with 1 vertex then there can be no edges, which establishes the basis step. Suppose it is true that any tree with n vertices has $n - 1$ edges, and let T be a tree with $|V(T)| = n + 1$. By Proposition 11.2 there is a vertex $v \in V(T)$ that is a leaf. Then $T - v$ is a tree with n vertices, and by our inductive hypothesis $|E(T - v)| = n - 1$. Now, because v is a leaf, removing v from T also removes precisely one edge from T, so that $|E(T)| = |E(T - v)| + 1 = n$. Having shown that any tree with $n + 1$ vertices has n edges, the proof is done. ∎

Corollary 11.4. *If G is a forest with n vertices and k components, then $|E(G)| = n - k$.*

Proof. Suppose G is a forest with $|V(G)| = n$ and k components. For each $1 \leq i \leq k$ let n_i be the number of vertices in component i. Since each component is a tree, by Proposition 11.3 the ith component has $n_i - 1$ edges. Therefore the total number of edges in G is

$$\sum_{i=1}^{k}(n_i - 1) = \sum_{i=1}^{k} n_i - k = n - k,$$

as claimed. ∎

Theorem 11.5. *Let T be a simple graph with $|V(T)| = n$. The following statements are equivalent.*

1. *T is a tree.*
2. *T is acyclic with $|E(T)| = n - 1$.*
3. *T is connected with $|E(T)| = n - 1$.*
4. *T is connected, and every edge is a cut-edge.*
5. *For every distinct $u, v \in V(T)$ there is exactly one u, v-path.*
6. *T is acyclic, and for any edge $e \notin E(T)$ the graph $T + e$ has exactly one cycle.*

Proof.
(1) → (2). Suppose T is a tree. Then T is acyclic by definition, with $|E(T)| = n - 1$ by Proposition 11.3.

(2) → (3). Suppose T is acyclic with $|E(T)| = n - 1$. Since T is a forest with n vertices, by Corollary 11.15 we find T cannot have more than one component. Therefore T is connected.

(3) → (4). Suppose T is connected with $|E(T)| = n - 1$, and let $e \in E(T)$. We have $|V(T - e)| = n$ and $|E(T - e)| = n - 2$, and since $|E(T - e)| < n - 1$, Proposition 10.80 implies that $T - e$ is not connected, and therefore e is a cut-edge.

(4) → (5). Suppose T is connected. Suppose there exist distinct $u, v \in V(T)$ for which there is not exactly one u, v-path. There is at least one such path by Proposition 10.72, and so there must exist two different u, v-paths. A path being a trail, it follows by Theorem 10.34 that there is a cycle in T. By Proposition 10.78 no edge in the cycle can be a cut-edge, and therefore not every edge in T is a cut-edge.

(5) → (6). Suppose that for every distinct $u, v \in V(T)$ there is exactly one u, v-path. Assume there is a cycle C in T. The length of C must be at least three since, being a simple graph, T cannot have a loop or parallel edges. Thus for some $n \geq 3$ there are distinct vertices $v_1, \ldots, v_n$ such that $C = v_1 v_2 \cdots v_n v_1$, and we find that there are two distinct v_1, v_2-paths: $v_1 v_2$ and $v_1 v_n v_{n-1} \cdots v_2$. This being a contradiction, we conclude that T is acyclic.

For vertices $x, y \in V(T)$ suppose that $e = xy$ is not an edge in $E(T)$. If $x = y$, then e is a loop at x and $T + e$ has exactly one cycle, so we assume that $x \neq y$. By hypothesis T

has exactly one x, y-path P_1 with vertex sequence $xv_1 \cdots v_k y$ for some $k \geq 1$, which does not traverse edge e. The graph $T + e$ has two x, y-paths: P_1, and also the path with vertex sequence xy, and hence $T + e$ contains the cycle $xv_1 \cdots v_k yx$. Suppose $T + e$ contains more than one cycle. These cycles must traverse e since T is acyclic, so that the differences between the cycles can only involve edges in T. But then if e is deleted from the cycles the result will be two or more distinct x, y-paths in T—a contradiction. Therefore $T + e$ has exactly one cycle.

(6) → (1). Let T be acyclic, and suppose T is not a tree. Then T is disconnected, with components H_1 and H_2. For $u \in H_1$ and $v \in H_2$, add the edge $e = uv$ to T. Suppose $T + e$ has a cycle C. Then C must necessarily traverse e, so that it may be characterized as having vertex sequence $uvw_1 \cdots w_k u$. This cycle must pass from H_1 to H_2 via e, and then back to H_1 via some edge $e' \neq e$. But any such edge e' would be a bridge connecting H_1 and H_2 in T, which is a contradiction. We conclude that $T + e$ is acylic; that is, there is an edge $e \notin E(T)$ such that $T + e$ does not have exactly one cycle. ■

As the next theorem shows, the number of leaves in a tree T may be expressed in terms of the number of vertices of T that are not leaves. For the theorem's statement we define the **maximal degree** of an undirected graph G, denoted by $\Delta(G)$, to be the largest degree attained by the vertices of G; that is,

$$\Delta(G) = \max\{d_G(v) : v \in V(G)\}.$$

Theorem 11.6. *Let T be a tree with $|V(T)| \geq 2$, and define $n_i = |\{v \in V(T) : d(v) = i\}|$ for $1 \leq i \leq \Delta(T)$. Then*

$$n_1 = 2 + \sum_{i=3}^{\Delta(T)} (i-2)n_i \tag{11.1}$$

Proof. Let $n = |V(T)|$, so that $|E(T)| = n - 1$ by Proposition 11.3. Now by Theorem 10.21,

$$2(n-1) = 2|E(T)| = \sum_{v \in V(T)} d(v) = \sum_{i=1}^{\Delta(T)} in_i,$$

and since $n = \sum_{i=1}^{\Delta(T)} n_i$, it follows that

$$2\sum_{i=1}^{\Delta(T)} n_i - 2 = \sum_{i=1}^{\Delta(T)} in_i,$$

which then yields (11.1). ■

Example 11.7. A tree T has exactly 22 leaves, four vertices of degree 4, and three vertices of degree 6. There are also exactly two vertices of degree ℓ. What is ℓ?

Solution. In the notation of Theorem 11.6 we are given that $n_1 = 22$, $n_4 = 4$, and $n_6 = 3$. Employing (11.1), we have

$$22 = 2 + n_3 + 2(4) + 3n_5 + 4(3) + 5n_7 + \cdots + (\Delta(T) - 2)n_{\Delta(T)}$$

whence comes

$$n_3 + 3n_5 + 5n_7 + \cdots + (\Delta(T) - 2)n_{\Delta(T)} = 0.$$

No term on the left side of this equation can be negatively valued, and so they must all equal 0. Thus $n_3 = n_5 = 0$, and also $n_k = 0$ for $k \geq 7$, which implies that $\Delta(T) = 6$. The only possibility left is for the two remaining vertices in T to each have degree 2, and therefore $\ell = 2$. ■

Example 11.8. Show that no tree with 100 vertices can have vertices of only degree 1 and 5.

Solution. Suppose T is such a tree. Then $\Delta(T) = 5$ and, in the notation of Theorem 11.6, $n_i = 0$ for all $i \notin \{1, 5\}$. Now,

$$n_1 = 2 + \sum_{i=3}^{5}(i-2)n_i = 2 + 3n_5,$$

and since $n_1 + n_5 = 100$, it follows that $n_1 = 2 + 3(100 - n_1)$ and hence $n_1 = 75.5$. This being impossible, there can exist no such tree as T. ■

Recalling that $\delta(G)$ denotes the minimum degree attained by the vertices of a graph G, the following theorem furnishes a simple criterion that a simple graph may satisfy to ensure that any tree with n vertices is isomorphic to a subgraph of that graph.

Theorem 11.9. *Let T be a tree and G a simple graph. If $|V(T)| = n \geq 1$ and $\delta(G) \geq n - 1$, then T is isomorphic to a subgraph of G.*

Proof. Suppose $|V(T)| = 1$ and $\delta(G) \geq 0$. Then T consists of a single vertex and G has at least one vertex (otherwise $\delta(G)$ would be undefined), and so T is isomorphic to any subgraph of G that has one vertex and no edges. The conditional statement (C) of the theorem is true when $n = 1$.

Next suppose $|V(T)| = 2$ and $\delta(G) \geq 1$. Then T consists of precisely two vertices joined by a single edge, and there exists some vertex u in G that is adjacent to at least one other vertex v by a single edge e. Clearly $T \simeq (\{u, v\}, \{e\})$, and so (C) is true when $n = 2$.

Assume the conditional statement of the theorem is true for some $n \geq 2$, and suppose $|V(T)| = n + 1$ and $\delta(G) \geq n$. Then T has at least one edge by Proposition 11.3, and so has a leaf v by Proposition 11.2. Seeing as $T - v$ is a tree with $|V(T - v)| = n$ and $\delta(G) \geq n - 1$, our inductive hypothesis implies that there is a subgraph H of G such that $T - v \simeq H$. Let $\varphi : V(T - v) \to H$ be an isomorphism, and let u denote the sole vertex joined to v in T, so that $uv \in E(T)$. We have $u \in V(T - v)$, so that $\varphi(u) \in V(H) \subseteq V(G)$ with $d_H(\varphi(u)) \leq n - 1$ (since $|V(H)| = n$) and $d_G(\varphi(u)) \geq n$. This implies there is some $w \in V(G) - V(H)$ for which $\varphi(u)w \in E(G)$. Now $(H \cup \{w\}) + \varphi(u)w$ is a subgraph of G that is isomorphic to T. ■

Definition 11.10. *Let G be a graph. The* ***eccentricity*** *of a vertex $u \in V(G)$, denoted by $\epsilon(u)$ or $\epsilon_G(u)$, is the distance from u to the vertex $v \in V(G)$ that is farthest from u, and thus*

$$\epsilon_G(u) = \max_{v \in V(G)} d_G(u, v).$$

A ***central vertex*** *of G is a vertex with minimal eccentricity, with C_G denoting the set of all central vertices of G. The* ***center*** *of G, denoted by Z_G, is the subgraph induced by C_G.*

Lemma 11.11. *Let T be a tree with $|V(T)| \geq 3$.*
1. *If u is a leaf of T and w is its neighbor, then $\epsilon(u) = \epsilon(w) + 1$.*
2. *If u is a central vertex of T, then $d(u) \geq 2$.*

Proof.
Proof of (1). Suppose $d(u) = 1$ and $w \in N(u)$. Let $v \in V(T)$ with $v \neq u, w$ be arbitrary. Any u, v-path of minimal length must start by traversing edge uw, and so $d(u, v) = d(w, v) + 1$. Moreover, $d(u, v) \geq 2 > 1 = d(u, w) > 0 = d(u, u)$, so that

$$\max_{v \neq u,w} d(u, v) = \max_{v \in V(T)} d(u, v),$$

and hence

$$\epsilon(u) = \max_{v \in V(T)} d(u, v) = \max_{v \neq u,w} d(u, v) = \max_{v \neq u,w} [d(w, v) + 1] = \epsilon(w) + 1.$$

Proof of (2). Suppose $d(u) < 2$. Since T is connected it follows that $d(u) = 1$, and so u has a neighbor w. Now, $\epsilon(u) > \epsilon(w)$ by part (a), and therefore u is not a central vertex of T. ■

Lemma 11.12. *Let T be a tree. If $u, w \in V(T)$ are such that $\epsilon(u) = d(u, w)$, then w is a leaf.*

Proof. Fix $u, w \in V(T)$ and suppose w is not a leaf, so that $d(w) \geq 2$. Let x and y be neighbors of w. By Theorem 11.5 there is a unique u, w-path P, and it cannot pass through both x and y. Indeed, if P did such a thing, then it would have a vertex sequence such as $u \cdots x \cdots y \cdots w$ (with the possibility that $u = x$ if $u \in N(w)$), so that $u \cdots xw$ would be another u, w-path distinct from P—a contradiction. We henceforth assume that P passes through x but not y, so that $P = u \cdots xw$ and P has length $d(u, w)$.

Now, we claim that there can exist no u, y-path that does not pass through w. To see this, we observe that if such a path existed, then it would not be a unique u, y-path since P may be extended to become the u, y-path $u \cdots xwy$ that does pass through w—again contradicting Theorem 11.5. Hence the only possible u, y-path is the aforementioned extension of P to y, so that $d(u, y) = d(u, w) + 1$, and then

$$\epsilon(u) = \max_{v \in V(T)} d(u, v) \geq d(u, y) > d(u, w).$$

Therefore $\epsilon(u) \neq d(u, w)$. ■

Lemma 11.13. *Let T be a tree with $|V(T)| \geq 3$, let $V' = \{v \in V(T) : d(v) = 1\}$, and let $T' = T - V'$.*
1. *T' is a tree.*
2. *If $u \in V(T')$, then $\epsilon_T(u) = \epsilon_{T'}(u) + 1$.*

Proof.
Proof of (1). Since T is acyclic and T' is a subgraph of T, it follows that T' is acyclic. Also, each $v \in V'$ is certainly the endpoint of a cut-edge of T, but because $d(v) = 1$ we conclude that v is not a cut-vertex of T by Proposition 10.79.

Proof of (2). Fix $u \in V(T')$, so u is not a leaf of T. Let w be any vertex in T such that $\epsilon_T(u) = d_T(u, w)$. Then w is a leaf of T by Lemma 11.12. Let x be the neighbor of w, which cannot be a leaf since $|V(T)| \geq 3$, and so $x \in V(T')$. There is a unique path P in T from u to w, which has length $\epsilon_T(u)$ and vertex sequence $u \cdots xw$. The unique u, x-path P' in T (which also exists and is unique in the tree T') traces P up to x and so has length $\epsilon_T(u) - 1$.

So, every $w \in V(T)$ with $d_T(u, w) = \epsilon_T(u)$ is a leaf of T, and upon deleting these leaves to obtain T' we find that every vertex $x \in V(T')$ that is of maximal distance from u is such that $d_{T'}(u, x) = \epsilon_T(u) - 1$, and so $\epsilon_{T'}(u) = \epsilon_T(u) - 1$. ■

The graph T' in Lemma 11.13 is an example of a **subtree**, which is any subgraph of a tree that is itself a tree.

The lemmas above may be used to prove a useful fact: stripping the leaves from a tree T having at least three vertices does not affect the center Z_T of the tree.

Proposition 11.14. *Let T be a tree with $|V(T)| \geq 3$, and let $V' = \{v \in V(T) : d(v) = 1\}$. If $T' = T - V'$, then $C_{T'} = C_T$.*

Proof. Suppose $T' = T - V'$. We begin by observing that if $x \in V'$, then x is a leaf of T and so cannot have minimal eccentricity by Lemma 11.11. Therefore, since $V(T) = V(T') \cup V'$, we have

$$\min_{v \in V(T')} \epsilon_T(v) = \min_{v \in V(T)} \epsilon_T(v).$$

Let $u \in C_T$, so that $\epsilon_T(u) = \min\{\epsilon_T(v) : v \in V(T)\}$. Then $d_T(u) \geq 2$ by Lemma 11.11, which implies $u \notin V'$ and hence $u \in V(T')$. Since $\epsilon_{T'}(v) = \epsilon_T(v) - 1$ for all $v \in V(T')$ by Lemma 11.13, we have

$$\epsilon_{T'}(u) = \epsilon_T(u) - 1 = \min_{v \in V(T)} [\epsilon_T(v) - 1] = \min_{v \in V(T')} [\epsilon_T(v) - 1] = \min_{v \in V(T')} \epsilon_{T'}(v),$$

and therefore $u \in C_{T'}$.

Next let $u \in C_{T'}$. Then certainly $u \in V(T')$, and by Lemma 11.13 we have

$$\epsilon_T(u) = \epsilon_{T'}(u) + 1 = \min_{v \in V(T')} [\epsilon_{T'}(v) + 1] = \min_{v \in V(T')} \epsilon_T(v) = \min_{v \in V(T)} \epsilon_T(v),$$

and therefore $u \in C_T$. ■

Corollary 11.15. *If T is a tree with at least three vertices and T' is the subgraph obtained by deleting the leaves of T, then $Z_{T'} = Z_T$.*

Proof. By Proposition 11.14 the graphs $Z_{T'}$ and Z_T each have vertex set C_T, and moreover each have edge set consisting of those edges in $E(T)$ whose endpoints both lie in C_T. Indeed, the only edges deleted in passing from T to T' are those incident with a leaf of T, and by Lemma 11.11 no leaf of T lies in C_T. ■

Corollary 11.16. *If T is a tree, then Z_T has either a single vertex or a single edge.*

Proof. We show using strong induction that, for all $n \geq 1$, if T is a tree with $|V(T)| = n$, then $|V(Z_T)| = 1$ or $|E(Z_T)| = 1$. The conditional statement $P(n)$ is true in the case when $n = 1$, since then $Z_T = T$ and so $|V(Z_T)| = |V(T)| = 1$. Fix $n \geq 1$, and suppose $P(k)$ is true for $1 \leq k \leq n$. Suppose T to be a tree with $|V(T)| = n + 1$. If $n = 1$, then $P(n + 1)$ follows since T must consist of two vertices joined by a single edge, and hence Z_T either consists of one vertex (so that $|V(Z_T)| = 1$) or two vertices (so that $|E(Z_T)| = 1$). We assume therefore that $n \geq 2$, which implies $|V(T)| \geq 3$. Then $|E(T)| \geq 2$ by Proposition 11.3, so that T has at least two leaves by Proposition 11.2, and thus the subtree T' as defined in Proposition 11.14 has at most $n-1$ vertices. Now, since $P(n-1)$ is true by hypothesis, we have either $|V(Z_{T'})| = 1$ or $|E(Z_{T'})| = 1$. However, $Z_T = Z_{T'}$ by Corollary 11.15, so that we have either $|V(Z_T)| = 1$ or $|E(Z_T)| = 1$, and therefore $P(n+1)$ is true. ■

11.2 – Rooted and Spanning Trees

When a tree sprouts from a consideration of some practical problem, there is often a vertex r in the tree that is of particular interest. Typically it is desirable to know the distance from r to each of the other vertices in the tree.

Definition 11.17. *A vertex in a tree that is designated to be a point of reference for all other vertices in the tree is called the tree's* ***root****. A* ***rooted tree*** *is a tree that has a root r, in which case we say the tree is* ***rooted at*** *r.*

Being a graph, every tree has a drawing. Once a vertex r is chosen to be the root of a tree, however, there are two prescribed ways in which the rooted tree is usually drawn. One way, which we could call the "top-down" approach, places the root r at the top of the drawing, with all vertices a distance 1 from r arranged in a row directly below r, all vertices a distance 2 from r in another row directly below the first one, and so on, so that all vertices a distance k from r are placed in a row just below the one formed by the vertices a distance $k-1$ from r. The vertices a distance k from r are said to be at **level** k, with r itself being at level 0. See Figure 24. The other common way of presenting a drawing of a rooted tree, the "left-right" approach, amounts to a top-down drawing rotated counterclockwise by 90°, so that the root r is placed leftmost, level 1 is a column of vertices to the right of r, and so on.

Any vertex in a tree may be designated to be the tree's root, thereby giving rise to a rooted tree. It may be wondered what, in specific mathematical terms, distinguishes one rooted tree with root r_1 from another with root r_2 if they both derive from the same tree T. Indeed, the manner in which we have defined a rooted tree above would result in the two aforementioned rooted trees being isomorphic! It must be admitted that our definition of a rooted tree is rather informal, as we will have no need for the formal definition. The formal definition adds a bit more structure: a rooted tree is a digraph whose underlying graph is a tree with a designated root r, and every arc is directed toward whichever of its two endpoints is more distant from the root. Thus if (u, v) is an arc in a rooted tree with root r, then $d(r, v) = d(r, u) + 1$. Hereafter we shall make no more mention of this formal characterization of rooted trees.

As in a family tree there are some terms that are commonly used to describe certain relationships between vertices in a rooted tree T with root r. Let $u, v \in V(T)$. If vertices u and v are adjacent, with u belonging to level k and v to level $k+1$, then v is called a **child** of u,

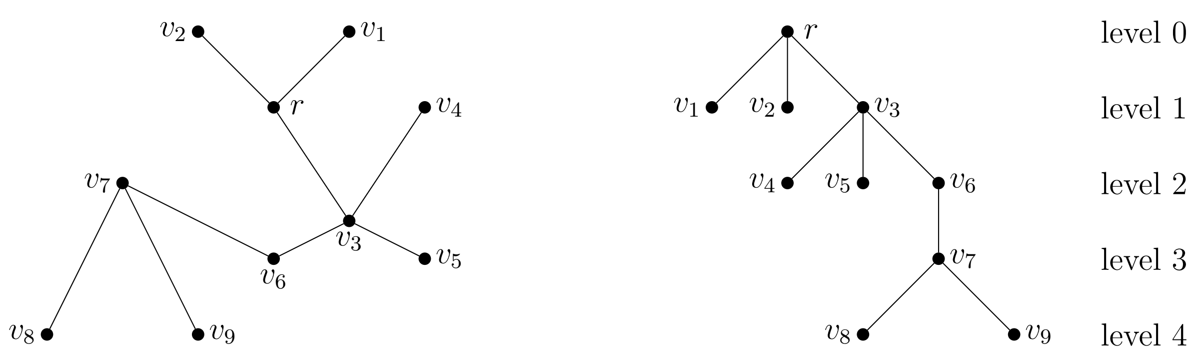


Figure 24. A rooted tree T with root r, drawn at right in the "top-down" format.

and u is the **parent** of v. Next, recalling from Theorem 11.5 that there is always precisely one path between any two vertices in a tree, we say v is a **descendant** of u (and u an **ancestor** of v) if u lies on the unique rv-path in T. Vertices having the same parent are **siblings**.

12

Further Topics in Graph Theory

12.1 – Matchings

We turn define the notion of a matching in graph theory, and establish a few results concerning it that involve bipartite graphs.

Definition 12.1. *Let $G = (V, E)$ be a simpled undirected graph. A **matching** in G is a set $M \subseteq E$ such that no two distinct edges in M have an endpoint in common. We say $v \in V$ is **matched** in M if it is an endpoint of some $e \in M$, otherwise it is **unmatched**. If $|N| \leq |M|$ for any matching N in G, then M is a **maximal matching**.*

Put another way, if $\{u_1, v_1\}$ and $\{u_2, v_2\}$ are two distinct edges belonging to a matching M in a simple undirected graph G, then the vertices u_1, u_2, v_1, v_2 are all distinct.

If $G = (V, E)$ is a bipartite graph with bipartition (V_1, V_2), we say a matching M in G is a **complete matching from** V_1 **to** V_2 if every $v \in V_1$ is an endpoint of some $e \in M$.

Proposition 12.2. *Let $G = (V, E)$ be a bipartite graph with bipartition (V_1, V_2). If M is a complete matching from V_1 to V_2, then $|M| = |V_1|$.*

Proof. Suppose that $M \subseteq E$ is a complete matching from V_1 to V_2. Since every vertex in V_1 is an endpoint of some edge in M, it must be that $|M| \geq |V_1|$. Now, if $|M| > |V_1|$ were the case, then there would be some edge $e \in M$ without an endpoint in V_1. But this is impossible: (V_1, V_2) being a bipartition, every edge in E (and hence in M) must have one endpoint in V_1 and another in V_2. Therefore $|M| = |V_1|$. ∎

Theorem 12.3 (Hall's Marriage Theorem). *If $G = (V, E)$ is a bipartite graph with bipartition (V_1, V_2), then G has a complete matching from V_1 to V_2 if and only if $|N_G(A)| \geq |A|$ for all $A \subseteq V_1$.*

Proof. Suppose $G = (V, E)$ is a bipartite graph (bigraph) with bipartition (V_1, V_2), and G has a complete matching M from V_1 to V_2. Let $A \subseteq V_1$. If $|A| = 0$, so that $A = \varnothing$, then $|N_G(A)| = 0 \geq 0 = |A|$. Assume $|A| = n$ for some $n \geq 1$, so that $A = \{a_1, \ldots, a_n\}$ with $a_i \neq a_j$

whenever $i \neq j$. Now, for each $1 \leq i \leq n$ there exists some $b_i \in V_2$ such that $\{a_i, b_i\} \in M$, and since $b_i \neq b_j$ whenever $i \neq j$, we find that $\{b_1, \ldots, b_n\} \subseteq N_G(A)$ and hence $|N_G(A)| \geq n = |A|$.

We prove the converse using strong induction on $|V_1|$. Specifically we prove $\forall n \geq 1[P(n)]$, where $P(n)$ is the statement "If $G = (V, E)$ is a bigraph with bipartition (V_1, V_2), $|N_G(A)| \geq |A|$ for all $A \subseteq V_1$, and $|V_1| = n$, then G has a complete matching from V_1 to V_2."

Suppose G is a bigraph with bipartition (V_1, V_2), $|N_G(A)| \geq |A|$ for all $A \subseteq V_1$, and $|V_1| = 1$. Setting $V_1 = \{u\}$, we have $|N_G(V_1)| \geq |V_1| = 1$, and so there exists some $v \in V_2$ such that $\{u, v\} \in E$. Letting $M = \{\{u, v\}\}$, it is clear that M is a complete matching for G from V_1 to V_2, which proves $P(1)$.

Fix $n \geq 1$, and suppose $P(i)$ for $1 \leq i \leq n$. Thus the inductive hypothesis (IH) states that if $H = (W, F)$ is a bigraph with bipartition (W_1, W_2), $|N_H(A)| \geq |A|$ for all $A \subseteq W_1$, and $|W_1| = i \leq n$, then H has a complete matching from W_1 to W_2. Suppose that $G = (V, E)$ is a bigraph with bipartition (V_1, V_2), $|N_G(A)| \geq |A|$ for all $A \subseteq V_1$, and $|V_1| = n + 1$. To prove G has a complete matching from V_1 to V_2 we consider two cases, the first case (C1) supposing that $|N_G(A)| \geq |A| + 1$ for any $A \subseteq V_1$ with $1 \leq |A| \leq n$. The second case (C2) will be the negation of (C1).

Suppose (C1). Fix $v \in V_1$. Since $|N_G(\{v\})| \geq |\{v\}| = 1$, there exists some $w \in V_2$ such that $\{v, w\} \in E$. Define $W = V - \{v, w\}$ and

$$F = \{\{x, y\} \in E : \{x, y\} \cap \{v, w\} = \varnothing\}.$$

Then $H = (W, F)$ is the graph that results when the vertices v and w are deleted from G, along with all edges incident to them. Also H is a bigraph with bipartition $(W_1, W_2) = (V_1 - \{v\}, V_2 - \{w\})$, with $|W_1| = n$.

Let $B \subseteq W_1$. If $B = \varnothing$ then $|N_H(B)| \geq |B|$ holds trivially, so assume that $1 \leq |B| \leq n$. Since $B \subseteq V_1$, the (C1) hypothesis implies that $|N_G(B)| \geq |B| + 1$. Now, because $v \notin B$, the removal of v from G, and all edges incident to v, does not by itself reduce the number of neighbors the vertices of B have; however, the removal of w from G may reduce the number of vertices neighboring B by at most one. Thus $N_H(B)$ equals either $N_G(B)$ or $N_G(B) - \{w\}$, so that $|N_H(B)| \geq |B|$ in any case, and therefore $|N_H(A)| \geq |A|$ for all $A \subseteq W_1$. By (IH) the graph H has a complete matching M from W_1 to W_2, and since no edge in M has v or w as an endpoint, it follows that G has complete matching $M \cup \{\{v, w\}\}$ from V_1 to V_2. This proves $P(n + 1)$ for case (C1).

Suppose (C2), so there exists some set $W_1 \subseteq V_1$ such that $1 \leq |W_1| \leq n$ and $|N_G(W_1)| < |W_1| + 1$. Since $|N_G(W_1)| \geq |W_1|$ as well, we have $|N_G(W_1)| = |W_1|$. Let $W_2 = N_G(W_1)$, $W = W_1 \cup W_2$, and $F = \{\{v, w\} \in E : v \in W_1\}$. Then $H = (W, F)$ is a bigraph with bipartition (W_1, W_2).

We now show that $N_H(A) = N_G(A)$ for any $A \subseteq W_1$. Let $w \in N_H(A)$, so there exists some $v \in A$ such that $\{v, w\} \in F$. Then $\{v, w\} \in E$ since $F \subseteq E$, so that $w \in N_G(A)$ and hence $N_H(A) \subseteq N_G(A)$. Next let $w \in N_G(A)$, so $\{v, w\} \in E$ for some $v \in A$. But then $v \in W_1$ since $A \subseteq W_1$, so that $\{v, w\} \in F$ and thus $w \in N_H(A)$. It follows that $N_G(A) \subseteq N_H(A)$, and therefore $N_H(A) = N_G(A)$. We use this fact to conclude that $|N_H(A)| = |N_G(A)| \geq |A|$ for all $A \subseteq W_1$, and hence (IH) implies that H has a complete matching M from W_1 to W_2.

Define $W_1' = V_1 - W_1$, $W_2' = V_2 - W_2$, $W' = W_1' \cup W_2'$, and

$$F' = \{\{v, w\} \in E : v \in W_1' \wedge w \in W_2'\},$$

so that $H' = (W', F')$ is a bigraph with bipartition (W_1', W_2'), and $|W_1'| = |V_1| - |W_1|$ implies that $1 \leq |W_1'| \leq n$. Suppose there is some $A \subseteq W_1'$ such that $|N_{H'}(A)| < |A|$. Now, since $A \cup W_1 \subseteq V_1$ and $A \cap W_1 = \varnothing$, we have

$$|N_G(A \cup W_1)| \geq |A \cup W_1| = |A| + |W_1|. \tag{12.1}$$

On the other hand, any vertices in $N_G(A) - N_{H'}(A)$ are in $V_2 - W_2' = W_2 = N_G(W_1)$, so that the number of neighbors of $A \cup W_1$ in G is the number of neighbors of A in H' plus the number of neighbors of W_1 in G. Recalling that $|N_G(W_1)| = |W_1|$, we obtain

$$|N_G(A \cup W_1)| = |N_{H'}(A)| + |N_G(W_1)| < |A| + |W_1|,$$

which contradicts (12.1). Therefore $|N_{H'}(A)| \geq |A|$ for all $A \subseteq W_1'$, and (IH) implies that H' has a complete matching M' from W_1' to W_2'. Then G has $M \cup M'$ as a complete matching from V_1 to V_2, thereby proving $P(n+1)$ for case (C2). ■

Hall's marriage theorem has a variety of applications, the proof of the next theorem being but one. For the statement of the theorem, given a bipartite graph G with bipartition (V_1, V_2), the **deficiency** of any $A \subseteq V_1$ is defined to be $\operatorname{def}(A) = |A| - |N_G(A)|$.

Theorem 12.4. *If $G = (V, E)$ is a bipartite graph with bipartition (V_1, V_2), then G has a matching in which at least $|V_1| - \max_{A \subseteq V_1} \operatorname{def}(A)$ vertices of V_1 are matched.*

Proof. Suppose $G = (V, E)$ is a bigraph with bipartition (V_1, V_2). Add $d := \max_{A \subseteq V_1} \operatorname{def}(A)$ new vertices to V_2, and join each of the new vertices to all the vertices in V_1. Letting D be the set of new vertices and $L = \{\{u, v\} : u \in V_1 \wedge v \in D\}$ the set of new edges, we obtain a new bigraph $H = (V \cup D, E \cup L)$ with bipartition $(V_1, V_2 \cup D)$. Since $|D| = d$, for any $B \subseteq V_1$ we have

$$\begin{aligned} |N_H(B)| = |N_G(B)| + |D| &= |N_G(B)| + \max_{A \subseteq V_1} \big(|A| - |N_G(A)|\big) \\ &\geq |N_G(B)| + \big(|B| - |N_G(B)|\big) = |B|, \end{aligned}$$

and so Hall's marriage theorem implies that H has a complete matching M' from V_1 to $V_2 \cup D$. Then $M := M' - L$ is a matching in G, and since $|M'| = |V_1|$ and M' has at most $|D|$ edges with endpoint in D, it follows that

$$|M| = |M' - L| \geq |M'| - |D| = |V_1| - \max_{A \subseteq V_1} \operatorname{def}(A).$$

Thus M is a matching in G in which at least $|V_1| - \max_{A \subseteq V_1} \operatorname{def}(A)$ vertices of V_1 are matched. ■

Example 12.5. There are four machines m_1, m_2, m_3, m_4, and each machine will be given one of four tasks t_1, t_2, t_3, t_4. Machine m_1 can do tasks t_1, t_3, t_4; machine m_2 can do tasks t_2, t_3; machine m_3 can do tasks t_3, t_4; and machine m_4 can do tasks t_1, t_2. Is it possible to assign tasks to the machines so that each machine is given exactly one task to do, and no task is left undone? If so, then find such an assignment of tasks.

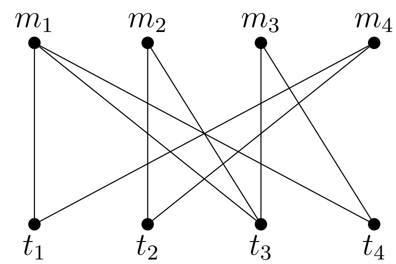


FIGURE 25.

Solution. In Figure 25 is shown a bipartite graph G that illustrates what tasks each machine can do, with $m_it_j \in E(G)$ if and only if machine m_i can do task t_j. A bipartition (V_1, V_2) of $V(G)$ has $V_1 = \{m_1, m_2, m_3, m_4\}$ and $V_2 = \{t_1, t_2, t_3, t_4\}$.

Let $A \subseteq V_1$. If $|A| = 1$, so that $A = \{m_i\}$ for some $1 \le i \le 4$. Every machine can do at least two tasks, and so

$$|N_G(A)| = |N_G(\{m_i\})| \ge 2 > 1 = |\{m_i\}| = |A|.$$

If $|A| = 2$, then $|N_G(A)| \ge 2 = |A|$; and if $|A| = 4$, then $A = V_1$ and $N_G(A) = V_2$, so that $|N_G(A)| = |A| = 4$. If $|A| = 3$ with $m_1 \in A$, then $|N_G(A)| \ge |N_G(\{m_3\})| = 3 = |A|$. Finally, if $|A| = 3$ with $m_1 \notin A$, then $A = \{m_2, m_3, m_4\}$, and by inspection we see that $|N_G(A)| = 4 > |A|$.

Thus $|N_G(A)| \ge |A|$, and by Hall's marriage theorem G has a complete matching from V_1 to V_2. This means there is a matching $M \subseteq E(G)$ such that each vertex in V_1 is an endpoint of some unique edge in M; that is, for each $m_i \in V_1$, there exists a unique $1 \le j \le 4$ such that $m_it_j \in M$. This means M designates a one-to-one correspondence between the elements of V_1 and V_2, so that each machine is given a single task to do, and no task is left undone.

One possible assignment of tasks is as follows: m_1 does t_3, m_2 does t_2, m_3 does t_4, and m_4 does t_1. ■

Example 12.6. In the distant future the clownfish and anemones in a particular reef have decided to systematize their ancient symbiotic relationship. In a poll it is found that each clownfish in the reef is willing to pair with any one of exactly 3 of the anemones in the reef, and each anemone in the reef is willing to pair with any one of exactly 3 of the clownfish. Also, a clownfish is willing to pair with an anemone if and only if the anemone is willing to pair with the clownfish. Show that it is possible to match the clownfish and anemones in the reef so that everyone is matched exclusively with someone with whom they are willing to be paired (as in a marriage).

Solution. Let m be the number of clownfish in the reef, and n the number of anemones. Naming the clownfish u_i for $1 \le i \le m$, and naming the anemones v_j for $1 \le j \le n$, and letting $V_1 = \{u_1, \dots, u_m\}$ and $V_2 = \{v_1, \dots, v_n\}$, we form a bipartite graph G whose vertex set $V(G)$ has bipartition (V_1, V_2). We have $uv \in E(G)$ if and only if $u \in V_1$ is willing to pair with $v \in V_2$.

Let $A \subseteq V_1$ with $|A| = p$. In order to show that $|N_G(A)| \ge |A|$, we consider the smallest value that $|N_G(B)|$ can attain for $B \subseteq V_1$ such that $|B| = p$. To minimize $|N_G(B)|$, we contrive for the elements of B to have as many neighbors in common as possible. Each element of B has 3 neighbors, and so the total number of neighbors of B is minimized if B is constructed from a disjoint union of sets, all of cardinality 3 except for possibly one set of cardinality less than 3, such that the elements belonging to a particular set all have the same neighbors. (It is not possible for 4 elements of B to have the same 3 neighbors, since that would mean there are 3 anemones that are each willing to pair with 4 clownfish.) By the division algorithm, $p = 3q + r$ for integers $q \geq 0$ and $0 \leq r \leq 2$. We have $b_1, b_2, b_3 \in B$ such that $N_G(b_1) = N_G(b_2) = N_G(b_3)$, and $b_4, b_5, b_6 \in B$ such that $N_G(b_4) = N_G(b_5) = N_G(b_6)$, and so on until arriving at $b_{3q-2}, b_{3q-1}, b_{3q} \in B$ such that $N_G(b_{3q-2}) = N_G(b_{3q-1}) = N_G(b_{3q})$. If $r \neq 0$, then there are one or two elements of B left over which also are taken to have the same set of neighbors, so that

$$N_G(B) = N_G(b_3) \sqcup N_G(b_6) \sqcup \cdots \sqcup N_G(b_{3q}) \sqcup N_G(b_{3q+r}),$$

and hence

$$|N_G(B)| = \sum_{i=1}^{q} |N_G(b_{3i})| + |N_G(b_{3q+r})| = \sum_{i=1}^{q} 3 + 3 = 3q + 3 > 3q + r = p.$$

It now follows that $|N_G(A)| \geq |N_G(B)| > p = |A|$.

By Hall's marriage theorem G has a complete matching from V_1 to V_2, so that every clownfish is matched with an anemone with whom it is willing to be paired in a symbiotic relationship. This shows that $m \leq n$; that is, there are at least as many anemones willing to enter into a relationship as clownfish. But an entirely symmetrical argument shows that G has a complete matching from V_2 to V_1, so that every anemone is matched with a clownfish with whom it is willing to be paired. This shows that $n \leq m$; that is, there are at least as many clownfish willing to enter into a relationship as anemones. Taken together, our findings establish that there are precisely as many clownfish willing to enter into a relationship as anemones, so that a complete matching from V_1 to V_2 is also a complete matching from V_2 to V_1. Therefore no one is left out, and everyone can be matched exclusively with someone with whom they are willing to be paired. ■

12.2 – Planar Graphs

The notion of a **drawing** of an undirected graph G has been with us since the beginning of Chapter 10. Thus far, however, we have paid no heed to questions of *how* G may be drawn, whether on a plane (such as a sheet of paper) or some other surface. Now we ask this question: Can a given graph G be drawn on a plane in such a way that no two of its edges cross each other? By "cross" is meant that the images (i.e. representations) of two edges in a drawing intersect at some point in the drawing that is not an image of a vertex. An instance of two edge images intersecting at a point that is not an image of an endpoint of either edge is called an **edge-crossing**.

Definition 12.7. *A **planar drawing** of a graph is a drawing of the graph in a plane without edge-crossings.*[12] *A graph is said to be **planar** if there exists a planar drawing of it, otherwise it is **nonplanar**.*

Example 12.8. In Figure 26(a) are two drawings of a graph G. The drawing at left is not a planar drawing since it features edge-crossings: edges ad and be cross each other, as well as edges ad and ce. The drawing at right in Figure 26(a), however, is a planar drawing, as it has no edge-crossings. Thus G is a planar graph.

At left in Figure 26(b) is a conventional drawing of the graph Q_3, the 3-cube, which features two edge-crossings. It is a planar graph, as demonstrated at right in Figure 26(b). ■

Definition 12.9. *A **Euclidean set** is any subset of a Euclidean space.*

Thus X is a Euclidean set if $X \subseteq \mathbb{R}^n$ for some $n \in \mathbb{Z}^+$. Henceforth, whenever $\mathbb{R}^n$ is mentioned, it will be understood that n may be any positive integer.

Definition 12.10. *Let $p, q \in \mathbb{R}^n$ with $p \neq q$, and let $C \subseteq \mathbb{R}^n$. If there exists a continuous bijection $\gamma : [0,1] \to \mathbb{R}^n$ such that $\gamma(0) = p$, $\gamma(1) = q$, and $\gamma([0,1]) = C$, then C is called an **open curve** from p to q.*

[12] Another term for a planar drawing of a graph is "plane graph," which causes no end of confusion owing to its similarity to "planar graph," and what is worse, the term gives the false impression that a drawing of a graph is the same thing as the graph itself.

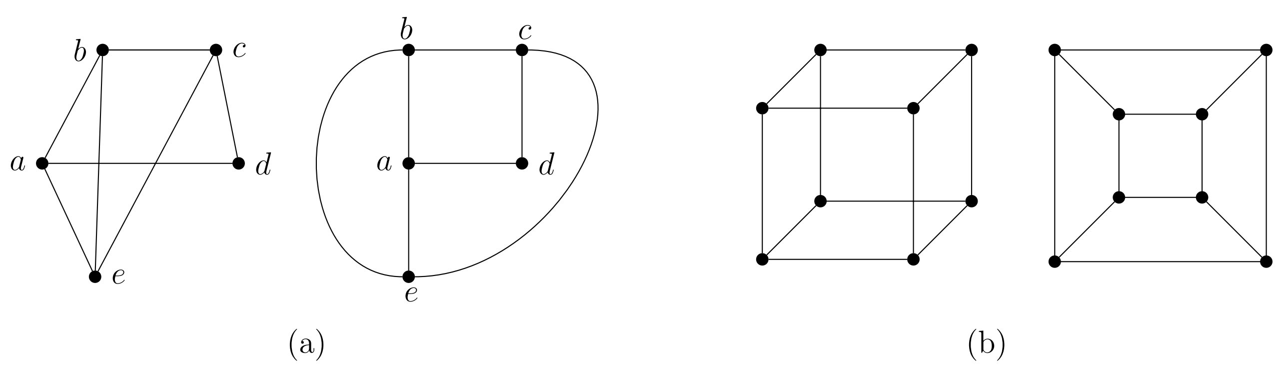


Figure 26.

Definition 12.11. *Let $p \in \mathbb{R}^n$ and $C \subseteq \mathbb{R}^n$. If there is a continuous function $\gamma : [0,1] \to \mathbb{R}^n$ such that $\gamma(0) = \gamma(1) = p$, $\gamma([0,1]) = C$, and $\gamma|_{(0,1)}$ is a bijection, then C is called a* ***closed curve*** *at p.*

If a curve C, whether open or closed, is equal to the set $\gamma([0,1])$, then the set $\gamma((0,1)) = C - \{\gamma(0), \gamma(1)\}$ is called the **interior** of C. Thus the interior of a curve is what is left when the curve is shorn of its starting and ending point.

A drawing of a graph G can be made on other surfaces besides a plane, such as a sphere or torus. If $S \subseteq \mathbb{R}^n$ is a surface, such as a plane, sphere, or torus, then a drawing of G on S is called an **embedding of G on S** if the drawing has no edge-crossings. A drawing of G is not the same thing as the graph G itself. A graph is a combinatorial construction, with objects (vertices) being "combined" by associations (edges) that join them, whereas a drawing of a graph is a topological construction consisting of sets of points on a surface. A connection between the two constructions (a graph G and surface S) is formally established by a **drawing function** $\mathfrak{e} : V(G) \cup E(G) \to \mathcal{P}(S)$ that maps each vertex $v \in V(G)$ to a set $\mathfrak{e}(v) \in \mathcal{P}(S)$ containing a single point in S, and each edge $e \in E(G)$ to a set of points $\mathfrak{e}(e) \in \mathcal{P}(S)$ that forms a curve in S.

The drawing function $\mathfrak{e} : V(G) \cup E(G) \to \mathcal{P}(S)$ is one-to-one on $V(G)$, so that $\mathfrak{e}(u) \neq \mathfrak{e}(v)$ whenever $u, v \in V(G)$ are such that $u \neq v$. Suppose $e \in E(G)$ has endpoints $u \neq v$. Then $\mathfrak{e}(u) = \{p_u\}$ and $\mathfrak{e}(v) = \{p_v\}$ for some points $p_u, p_v \in S$ such that $p_u \neq p_v$, and $\mathfrak{e}(e) = \gamma_e([0,1])$ for some open curve $\gamma_e : [0,1] \to S$ from p_u to p_v (so $\gamma_e(0) = p_u$ and $\gamma_e(1) = p_v$). If $u = v$, so that e is a loop at u, then $\mathfrak{e}(e) = \gamma_e([0,1])$ for some closed curve $\gamma_e : [0,1] \to S$ at p_u (so $\gamma_e(0) = \gamma_e(1) = p_u$). One last property that $\mathfrak{e}$ is required to satisfy is this: Given any $v \in V(G)$ and $e \in E(G)$, it must be that $\mathfrak{e}(v) \nsubseteq \gamma_e((0,1))$. That is, if $\mathfrak{e}(v) = \{p_v\}$, then $p_v \notin \gamma_e((0,1))$, and hence the image of no vertex in G lies in the interior of any curve that is the image of an edge in G.

Now, given a drawing function $\mathfrak{e} : V(G) \cup E(G) \to \mathcal{P}(S)$, we designate the set $\mathfrak{e}(G)$ defined by $\mathfrak{e}(G) = \bigcup \mathfrak{e}(V(G) \cup E(G))$ to be a **drawing of G on S**.[13] From the way in which the drawing function has been defined, we find that

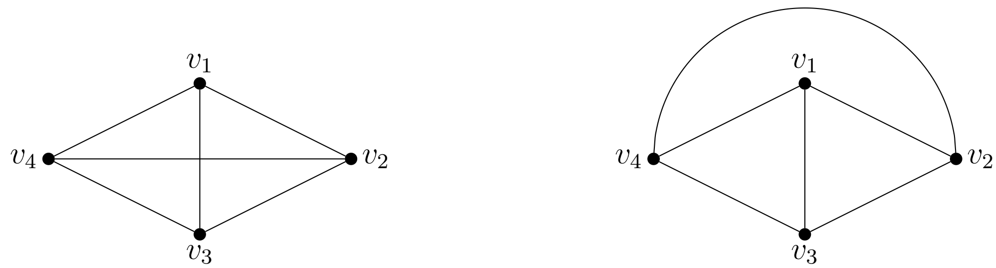
$$\mathfrak{e}(G) = \bigcup_{e \in E(G)} \gamma_e([0,1]). \tag{12.2}$$

If $\gamma_{e_1}((0,1)) \cap \gamma_{e_2}((0,1)) = \varnothing$ for all $e_1, e_2 \in E(G)$, then the drawing $\mathfrak{e}(G)$ features no curves that intersect at a point in S that is the image of a vertex in G, and hence the drawing $\mathfrak{e}(G)$ has no edge-crossings. Whenever $\mathfrak{e}(G)$ has no edge-crossings, then both the drawing function $\mathfrak{e} : V(G) \cup E(G) \to \mathcal{P}(S)$ and the drawing $\mathfrak{e}(G)$ are called an **embedding of G on S**. If $S = \mathbb{R}^2$, then $\mathfrak{e}$ and $\mathfrak{e}(G)$ are called a **planar embedding of** G, and indeed $\mathfrak{e}(G)$ in particular may also be called a **planar drawing** according to Definition 12.7.

Example 12.12. The graph K_4 has $V(K_4) = \{v_1, v_2, v_3, v_4\}$ and

$$E(K_4) = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4, v_3v_4\}.$$

[13]We emphasize that the symbols $\mathfrak{e}(G)$ and $\mathfrak{e}(V(G) \cup E(G))$ represent two different things: $\mathfrak{e}(G)$ is a set of *points in S* while $\mathfrak{e}(V(G) \cup E(G))$ is a set of *subsets of S*.



$\mathfrak{e}(K_4)$ $\mathfrak{e}'(K_4)$

FIGURE 27. Two drawings of K_4 in the plane, with coordinate axes suppressed.

Using the rectangular coordinate system on $\mathbb{R}^2$, we define a drawing function $\mathfrak{e} : V(K_4)\cup E(K_4) \to \mathcal{P}(\mathbb{R}^2)$ as follows:

$$\mathfrak{e}(v_1) = \{(0,1)\}, \quad \mathfrak{e}(v_2) = \{(2,0)\}, \quad \mathfrak{e}(v_3) = \{(0,-1)\}, \quad \mathfrak{e}(v_4) = \{(-2,0)\},$$

and $\mathfrak{e}(v_iv_j) = \gamma_{ij}([0,1])$, where $\gamma_{ij}(t) = (1-t)p_i + tp_j$ for $0 \le t \le 1$ and p_i the point for which $\{p_i\} = \mathfrak{e}(v_i)$. For example,

$$\gamma_{12}(t) = (1-t)p_1 + tp_2 = (1-t)(0,1) + t(2,0) = (2t, 1-t).$$

Thus the image of each edge v_iv_j in K_4 is a line segment in $\mathbb{R}^2$ (which is an open curve) from v_i to v_j. The drawing $\mathfrak{e}(K_4)$, shown at left in Figure 27, is not a planar embedding of K_4 since $\gamma_{24}((0,1)) \cap \gamma_{13}((0,1)) = \{(0,0)\}$; that is, the interiors of the curves given by γ_{24} and γ_{13} intersect at the point $(0,0)$, so there is an edge-crossing.

Now we define an alternative drawing function $\mathfrak{e}' : V(K_4) \cup E(K_4) \to \mathcal{P}(\mathbb{R}^2)$ as follows: $\mathfrak{e}'(x) = \mathfrak{e}(x)$ for all $x \in V(K_4) \cup E(K_4)$ such that $x \neq v_2v_4$, and $\mathfrak{e}'(v_2v_4) = \gamma'_{24}([0,1])$, where

$$\gamma'_{24}(t) = (2\cos(\pi t), 2\sin(\pi t)), \quad 0 \le t \le 1.$$

The drawing $\mathfrak{e}'(K_4)$, at right in Figure 27, is free of edge-crossings and therefore is a planar embedding of K_4. This shows that K_4 is a planar graph. ■

In practice, for $v \in V(G)$ we will call $\mathfrak{e}(v)$ a "point" (or even a "vertex"), and denote it by p_v rather than $\{p_v\}$; and for $e \in E(G)$ we will call $\mathfrak{e}(e)$ a "curve" (or even an "edge"), and denote it by a function $\gamma_e : [0,1] \to S$ whose range $\gamma_e([0,1])$ equals an appropriate set of points in S. (Informally we may let v denote $\mathfrak{e}(v)$ and e denote $\mathfrak{e}(e)$. See Remark 12.13 below.) The construction that we call the "drawing of G on S" we denote by $\mathfrak{e}(G)$ as we have already done. The drawing $\mathfrak{e}(G)$ is the union of all the sets in $\mathfrak{e}(V(G) \cup E(G))$, the image of $V(G) \cup E(G)$ under $\mathfrak{e}$. Until the end of the section the only surface S upon which we will be drawing graphs will be the Euclidean 2-space $\mathbb{R}^2$, which we refer to as "the plane." If $S = \mathbb{R}^2$, then a drawing $\mathfrak{e}(G)$ is an embedding if and only if it is a planar drawing.

Remark 12.13. Just as a drawing of a graph G is strictly a topological representation of G, and not G itself, so too the curves and points in the drawing are topological representations of the edges and vertices of G, respectively, and not the edges and vertices themselves. That is, the curves and points in the drawing are *images* of the edges and vertices in the graph.

These images are formed by a drawing function $\mathfrak{e}$, which may or may not be an embedding. Nonetheless we shall often (but not always) informally call these images "edges" and "vertices," rather than "curves" and "points," as has already been our practice since taking up the study of graph theory. We shall also often (but not always) informally let the same symbol that denotes a vertex or edge to also denote the image of the vertex or edge. ■

Given an embedding $\mathfrak{e} : V(G) \cup E(G) \to \mathcal{P}(\mathbb{R}^2)$ of a graph G on the plane $\mathbb{R}^2$, we will be interested in the anatomy of the set $\mathbb{R}^2 - \mathfrak{e}(G)$, which is what's left of the plane when the curves and points of the drawing $\mathfrak{e}(G)$ are removed. A series of definitions will be necessary to make our ideas precise.

Definition 12.14. *A Euclidean set X is* ***connected*** *if it consists of a single point, or if for every $p, q \in X$ with $p \neq q$ there exists an open path C from p to q such that $C \subseteq X$. A nonempty Euclidean set that is both open and connected is called a* ***region****.*

If S is a Euclidean set in $\mathbb{R}^n$, then $R \subseteq S$ is a ***region in*** *S if R is connected and there exists an open set U in $\mathbb{R}^n$ such that $R = S \cap U$.*

If $X_1, \ldots, X_m \subseteq \mathbb{R}^n$ are mutually disjoint regions, and $X = X_1 \cup \cdots \cup X_m$, then each X_i is a ***component*** *of X.*

The ***boundary*** *of an open connected set $X \subseteq \mathbb{R}^n$, denoted by ∂X, consists of all points $p \in \mathbb{R}^n - X$ for which every open set containing p also contains a point in X.*

This definition of the word "connected" is not to be confused with the notion of connectedness as it applies to graphs, whether undirected or directed. There is a correspondence between the two concepts, however: Any drawing of a connected graph, whether an embedding or not, will necessarily be a connected Euclidean set in $\mathbb{R}^2$. Our definition of the component of a Euclidean set X pertains only to open sets, which is all that we shall need aside from this fact: if $X, Y \subseteq \mathbb{R}^n$ are disjoint regions, then the set $X \cup Y$ is not connected (we say **disconnected**) since no open path from p to q exists if $p \in X$ and $q \in Y$.[14] An easy example is furnished by the open intervals $(0, 1)$ and $(1, 2)$ in $\mathbb{R}^1$ (which is $\mathbb{R}$): If $p \in (0, 1)$ and $q \in (1, 2)$, then there is no open path from p to q that lies entirely in $(0, 1) \cup (1, 2)$, as such a path must include all the points in the interval $[p, q]$, including 1.

Now let $\mathfrak{e}$ be a planar embedding for a graph G, which is to say the set $\mathfrak{e}(G)$ is a planar drawing of G in $\mathbb{R}^2$. The complement of the drawing, $\mathbb{R}^2 - \mathfrak{e}(G)$, will be comprised of disjoint regions $R_1, \ldots, R_m \subseteq \mathbb{R}^2$ for some $m \in \mathbb{Z}^+$. For each $1 \leq i \leq m$ we call R_i a **face** of the planar drawing $\mathfrak{e}(G)$. Thus any planar drawing of G has vertices, edges, and faces, analogous to any polyhedron.

A Euclidean set in $\mathbb{R}^2$ is **bounded** if it is a subset of an open disc of finite radius, otherwise it is **unbounded**. If $\mathfrak{e}(G)$ is a planar embedding of G, then the set $\mathbb{R}^2 - \mathfrak{e}(G)$ will always have precisely one unbounded region, called the **exterior face** of $\mathfrak{e}(G)$, and any other regions will be bounded and called **interior faces**.

Example 12.15. At left in Figure 28 is a planar drawing of a graph G. Being embedded in the plane $\mathbb{R}^2$, which is shaded, the drawing $\mathfrak{e}(G)$ constitutes a Euclidean set.

At center in the figure is shown $\mathbb{R}^2 - \mathfrak{e}(G)$. The deletion of the drawing from the plane disconnects the latter into four faces: f_1, f_2, f_3, and f_4. All faces are interior faces save for f_4,

[14] The tools to prove such a fact would be given in a textbook such as *Topology* by James Munkres.

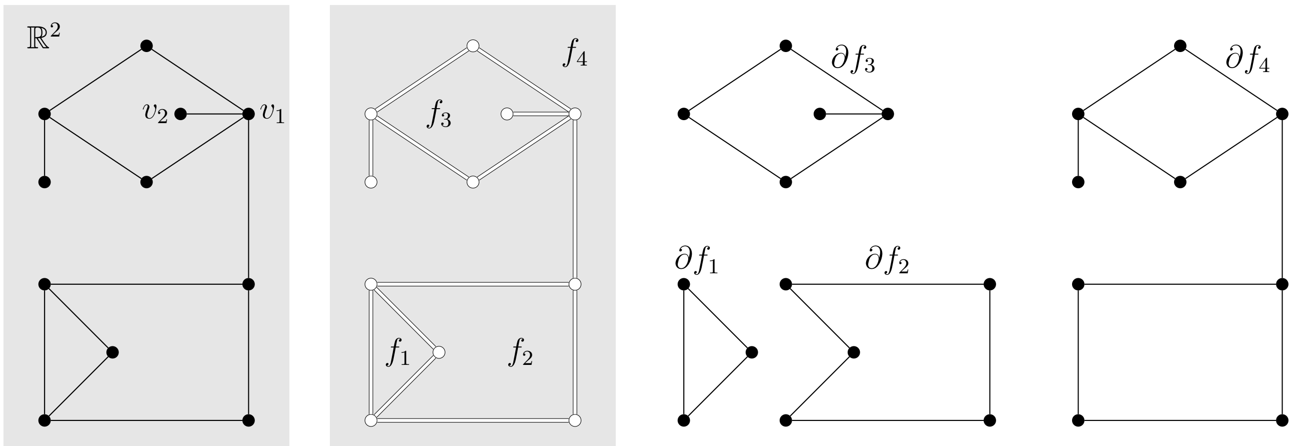


FIGURE 28.

which is unbounded and is therefore the exterior face. Being open sets, none of the faces contain any point that lies in the drawing $\mathfrak{e}(G)$.

Finally, at right in the figure are shown the boundaries of the four faces of $\mathfrak{e}(G)$. We observe that though f_4 is an unbounded region, it nonetheless possesses a boundary! ■

We shall soon see that the number of faces a planar drawing of a planar graph $G = (V, E)$ possesses does not depend on the choice of embedding $\mathfrak{e}$. Indeed, the number of faces will always equal $2 - |V| + |E|$. To prove this, however, we need a couple theoretical results.

Proposition 12.16. *Every subgraph of a planar graph is planar.*

Proof. Let G be a planar graph. Any subgraph H of G may be obtained by deleting vertices and edges from G, and correspondingly a drawing of H may be obtained by deleting the images of the same vertices and edges from a planar drawing of G. Since deleting edges from a drawing without edge-crossings can only produce a new drawing that is likewise devoid of edge-crossings, the resultant drawing of H must be a planar drawing, and therefore H is planar. ■

If H is a subgraph of a planar graph G, and $\mathfrak{e}(G)$ is a planar drawing of G, then $\mathfrak{e}(H)$ is the planar drawing of H that is obtained by deleting from the drawing $\mathfrak{e}(G)$ the images of precisely those edges and vertices in G that must be removed to obtain the subgraph H. In other words, the drawing $\mathfrak{e}(H)$ is what results by restricting the domain of the embedding $\mathfrak{e} : V(G) \cup E(G) \to \mathcal{P}(\mathbb{R}^2)$ to $V(H) \cup E(H)$; that is, to the vertices and edges of G that are contained in H.

Lemma 12.17. *Let G be a simple connected planar graph with planar drawing $\mathfrak{e}(G)$. Suppose $u, v \in V(G)$ are distinct vertices such that $uv \notin E(G)$. If there does not exist a face f of $\mathfrak{e}(G)$ such that $u, v \in \partial f$,*[15] *then $G + uv$ is nonplanar.*

[15]Strictly speaking u and v are points in $\mathbb{R}^2$ here, not vertices in G, and we should write $\mathfrak{e}(u), \mathfrak{e}(v) \in \partial f$. But our practice has nearly always been to use the same symbol for both a vertex in a graph and its image on the plane. See Remark 12.13.

Proof. In accord with our definitions, a face of the drawing $\mathfrak{e}(G)$ is a region of $\mathbb{R}^2 - \mathfrak{e}(G)$. Suppose there is no region of $\mathbb{R}^2 - \mathfrak{e}(G)$ whose boundary contains both u and v. We now purpose to add a new edge uv to the planar drawing $\mathfrak{e}(G)$ of graph G. Starting at u, such an edge must first enter a face f for which $u \in \partial f$, and since $v \notin \partial f$, the edge must next pass into at least one other face f' before reaching v. Let $p \in f$ and $q \in f'$ be points lying on uv. Since $f \cup f'$ is disconnected, the open path from p to q along edge uv is not a subset of $f \cup f'$, and so there is a point x on uv that lies outside $f \cup f'$. Not lying in any other region of $\mathbb{R}^2 - \mathfrak{e}(G)$, the point x must lie in $\mathfrak{e}(G)$; and since uv cannot contain any vertices in G other than u and v, the point x on uv must lie on some other edge e in the drawing $\mathfrak{e}(G)$, but not at an endpoint of e. Thus uv crosses e, and we conclude that any drawing of $G + uv$ necessarily possesses at least one edge-crossing. Therefore $G + uv$ is nonplanar. ■

Let G be a planar graph. For the statement of the following theorem and several later results, we shall denote by $F(G)$ the set of faces of a planar drawing $\mathfrak{e}(G)$.

Theorem 12.18 (The Euler Identity). *If G is a simple connected planar graph, then*

$$|V(G)| - |E(G)| + |F(G)| = 2.$$

Proof. Suppose G is a simple connected planar graph with planar drawing $\mathfrak{e}(G)$. It will be convenient to let $v = |V(G)|$, $e = |E(G)|$, and $f = |F(G)|$. The proof depends on constructing a sequence $G_1, G_2, \dots, G_e$ of subgraphs of G, with G_1 being an arbitrarily chosen subgraph with one edge, and G_{k+1} being constructed from G_k by adding an edge of G that is incident to at least one vertex in G_k. Thus $|E(G_k)| = k$ for each $1 \leq k \leq e$, with $G_e = G$. Since G is planar, Proposition 12.16 assures us that G_k is likewise planar for all $1 \leq k \leq e$. Define $v_k = |V(G_k)|$ and $e_k = |E(G_k)|$, and let f_k be the number of faces of $\mathfrak{e}(G_k)$. We prove by induction that $v_k - e_k + f_k = 2$ for all k.

The simple graph G_1 has one edge and two vertices, and hence the only face of the planar drawing $\mathfrak{e}(G_1)$ is $\mathbb{R}^2 - C$ for some open curve C. Thus we have $v_1 = 2$, $e_1 = 1$, and $f_1 = 1$, so that $v_1 - e_1 + f_1 = 2$ and the base case is affirmed.

Now let $k \geq 1$ be arbitrary and suppose $v_k - e_k + f_k = 2$. Choose an edge $x_{k+1}y_{k+1}$ to add to G_k to obtain G_{k+1}. There are two cases to consider: (I) $x_{k+1}, y_{k+1} \in V(G_k)$; and (II) $x_{k+1} \in V(G_k)$, $y_{k+1} \notin V(G_k)$. Suppose case (I) is the reality. Then the adding of $x_{k+1}y_{k+1}$ to G_k results in G_{k+1} having the same number of vertices as G_k, so that $v_{k+1} = v_k$ and $e_{k+1} = e_k + 1$. Since $G_{k+1} = G_k + x_{k+1}y_{k+1}$ is planar, by Lemma 12.17 there exists a face X of $\mathfrak{e}(G_k)$ such that $x_{k+1}, y_{k+1} \in \partial X$, and so the addition of edge $x_{k+1}y_{k+1}$ to G_k will result in X being split into two faces X_1 and X_2 of $\mathfrak{e}(G_{k+1})$. Hence $f_{k+1} = f_k + 1$, and

$$v_{k+1} - e_{k+1} + f_{k+1} = v_k - (e_k + 1) + (f_k + 1) = v_k - e_k + f_k = 2.$$

This proves the inductive step for case (I).

Now we turn to case (II). In this scenario $V(G_{k+1}) = V(G_k) \cup \{y_{k+1}\}$ and $E(G_{k+1}) = E(G_k) \cup \{x_{k+1}y_{k+1}\}$, so that $v_{k+1} = v_k + 1$ and $e_{k+1} = e_k + 1$. To determine f_{k+1}, we note that since $y_{k+1} \notin \mathfrak{e}(G_k)$, and hence $y_{k+1} \in \mathbb{R}^2 - \mathfrak{e}(G_k)$, it must be that y_{k+1} lies in a component X of $\mathbb{R}^2 - \mathfrak{e}(G_k)$. This component X is a face of $\mathfrak{e}(G_k)$, with $y_{k+1} \in D \subseteq X$ for some open disc D centered at y_{k+1}, and the embedding $\mathfrak{e}(G_{k+1})$ is constructed from $\mathfrak{e}(G_k)$ by drawing an open curve C from y_{k+1} in the interior of X to x_{k+1}. The curve C represents the edge $x_{k+1}y_{k+1}$, and we must have $x_{k+1} \in \partial X$ so as to avoid edge-crossings. (To have $x \in \mathbb{R}^2 - (X \cup \partial X)$ would force C to pass from the component X of $\mathbb{R}^2 - \mathfrak{e}(G_k)$ into some other component.) Now, since $d_{G_{k+1}}(y_{k+1}) = 1$, we find that $X' = X - C$ is connected. Indeed, X' may be regarded as being formed from X by deleting a line segment from a point on its boundary to some interior point, so that X' is the only face of $\mathfrak{e}(G_{k+1})$ that replaces face X of $\mathfrak{e}(G_k)$, and therefore $f_{k+1} = f_k$. Finally:

$$v_{k+1} - e_{k+1} + f_{k+1} = (v_k + 1) - (e_k + 1) + f_k = v_k - e_k + f_k = 2.$$

This proves the inductive step for case (II), and the theorem is proven. ■

Definition 12.19. *Let G be a graph and S a surface. The* ***boundary walk*** *of a face f of an embedding of G on S is the length of a shortest walk that traverses every edge of ∂f at least once. The length of a boundary walk of f is the* ***degree*** *of f, denoted by* $\deg(f)$.

Example 12.20. The graph G with $V(G) = \{u, v\}$ and $E(G) = uv$ is planar. A possible embedding is $\mathfrak{e} : V(G) \cup E(G) \to \mathcal{P}(\mathbb{R}^2)$ with $\mathfrak{e}(u) = (0,0)$, $\mathfrak{e}(v) = (1,0)$, and $\mathfrak{e}(uv)$ the line segment in the plane from point $(0,0)$ to $(1,0)$. Indeed, we could have $\mathfrak{e}(uv) = \gamma$, where $\gamma : [0,1] \to \mathbb{R}^2$ is given by $\gamma(t) = (t, 0)$ for all $0 \le t \le 1$. The planar drawing $\mathfrak{e}(G)$ has but one face: the external face $f = \mathbb{R}^2 - \{(t,0) : 0 \le t \le 1\}$ (which may be written $\mathbb{R}^2 - \gamma([0,1])$ or even $\mathbb{R}^2 - [0,1]$). This is a more "formally correct" characterization of an embedding, and the boundary of f is $\partial f = \{(t,0) : 0 \le t \le 1\}$.

Ratcheting down the formality, we may describe a planar drawing of G as consisting simply of the "vertices" u and v in the plane, joined by an "edge" $e = uv$. The external face f has boundary $\partial f = e$, which of course includes the endpoints u and v. There are only two possible boundary walks for f: $ueveu$ and $veuev$ (both of which have edge sequence ee). Having length 2, we conclude that the degree of f is $\deg(f) = 2$. ■

Proposition 12.21. *If G is a simple connected planar graph, then*

$$2|E(G)| = \sum_{f \in F(G)} \deg(f). \tag{12.3}$$

Proof. Let $e \in E(G)$. There are two cases: e lies on the boundary of two adjacent faces f_1 and f_2, or e lies on the boundary of only one face f.

In the former case, e appears once in the edge sequence of a boundary walk of f_1, and also once in the edge sequence of a boundary walk of f_2, so that e contributes 1 to each of $\deg(f_1)$ and $\deg(f_2)$, and hence contributes 2 to the sum at right in (12.3).

In the latter case, e must thrust into the face f, appearing as a "spike" in the midst of f rather like the edge v_1v_2 in face f_3 of Figure 28. Such an edge e will need to be traversed twice in a boundary walk, and so again contributes 2 to the sum in (12.3).

We conclude that every $e \in E(G)$ contributes 2 to $\sum_{f \in F(G)} \deg(f)$, thereby affirming that the sum equals twice the total number of edges. ■

Proposition 12.22. *If G is a simple connected planar graph and $|V(G)| \ge 3$, then* $\deg(f) \ge 3$ *for all $f \in F(G)$.*

Proof. Let $f \in F(G)$. To have $\deg(f) = 1$ is only possible if ∂f is a loop, which the simple graph G lacks. To have $\deg(f) = 2$ may be accomplished in two ways. One way is for ∂f to consist of two parallel edges, which again is precluded by the hypothesis that G is simple; and the other way is for f to be the external face in a planar drawing of the graph described in Example 12.20, which runs afoul of our hypothesis that G have at least three vertices. Indeed, the connected graph with the fewest edges permissible is $K_{1,2}$, for which the degree of the external face is 4; and adding another edge to $K_{1,2}$ to obtain the triangle K_3 results in two faces that each have degree 3. ■

With the theory we have developed thus far, we can now fashion some simple algebraic tests that may be used to determine the planarity or nonplanarity of countless different graphs.

Proposition 12.23. *If G is a simple connected planar graph with at least 3 vertices, then $|E(G)| \leq 3|V(G)| - 6$.*

Proof. Since $\deg(f) \geq 3$ for all $f \in F(G)$ by Proposition 12.22, with Proposition 12.21 we obtain

$$2|E(G)| = \sum_{f \in F(G)} \deg(f) \geq \sum_{f \in F(G)} 3 = 3|F(G)|,$$

and hence $|F(G)| \leq \frac{2}{3}|E(G)|$. Now, by Theorem 12.18,

$$|E(G)| = |V(G)| + |F(G)| - 2 \leq |V(G)| + \frac{2}{3}|E(G)| - 2$$

which yields $|E(G)| \leq 3|V(G)| - 6$ as was to be shown. ■

Proposition 12.24. *If G is a simple connected planar graph with at least 3 vertices and no triangles, then $|E(G)| \leq 2|V(G)| - 4$.*

Proof. By Proposition 12.22, $\deg(f) \geq 3$ for all $f \in F(G)$. However, the only face that can have degree 3 is one whose boundary is a triangle, and since G has no triangles by hypothesis, it must be that $\deg(f) \geq 4$ for all $f \in F(G)$. Now, by Proposition 12.21,

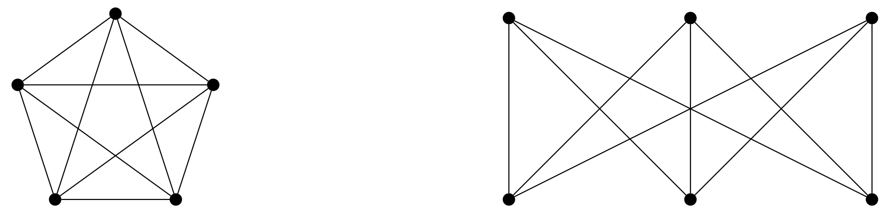
$$2|E(G)| = \sum_{f \in F(G)} \deg(f) \geq \sum_{f \in F(G)} 4 = 4|F(G)|,$$

and hence $|F(G)| \leq \frac{1}{2}|E(G)|$. By Theorem 12.18,

$$|E(G)| = |V(G)| + |F(G)| - 2 \leq |V(G)| + \frac{1}{2}|E(G)| - 2$$

which yields $|E(G)| \leq 2|V(G)| - 4$ as was to be shown. ■

There is the **utility problem**, which may be stated thus: Three houses are on a street, side-by-side, and they each need three utilities: gas, electricity, and water. The source for each of these utilities is located at a point distinct from the others, and so each house requires a gas line, electrical cable, and water pipe to join it directly to each of the three access points. Is there a way to do this without any lines, cables, or pipes crossing? In a graph G the houses are three vertices h_1, h_2, h_3, the utilities are three more vertices u_1, u_2, u_3, and each vertex h_i must be joined with each of the vertices u_1, u_2, u_3 to form the bipartite graph $K_{3,3}$. The utility problem



K_5 $\qquad$ $K_{3,3}$

FIGURE 29. The Kuratowski graphs.

is equivalent to asking whether graph $K_{3,3}$ has a planar drawing. Is $K_{3,3}$ a planar graph or not? A bit of trial and error may be enough to convince one that the answer is no, but how to prove it?

We now have the tools to show easily that the so-called **Kuratowski graphs**,[16] which are K_5 and $K_{3,3}$, are both nonplanar. See Figure 29.

Proposition 12.25. *The graphs K_5 and $K_{3,3}$ are nonplanar.*

Proof. For K_5 we have $|E(K_5)| = 10$ and $|V(K_5)| = 5$, and so $|E(K_5)| > 3|V(K_5)| - 6$. Since K_5 is clearly a simple connected graph with at least three vertices, by the contrapositive of Proposition 12.23 we conclude that K_5 is nonplanar.

For $K_{3,3}$ we have $|E(K_{3,3})| = 9$ and $|V(K_{3,3})| = 6$, and so $|E(K_{3,3})| > 2|V(K_{3,3})| - 4$. Since $K_{3,3}$ is seen to be a simple connected graph with at least three vertices and no triangles, by the contrapositive of Proposition 12.24 we conclude that $K_{3,3}$ is nonplanar. ∎

We observe that Proposition 12.23 is of no use in showing that $K_{3,3}$ is nonplanar, since $|E(K_{3,3})| = 9$ and $|V(K_{3,3})| = 6$ implies $|E(K_{3,3})| \leq 3|V(K_{3,3})| - 6$. Indeed, $K_{3,3}$ shows that the converse of Proposition 12.23 is not true in general; that is, if a simple connected graph G satisfies the inequality $|E(G)| \leq 3|V(G)| - 6$, then it does not necessarily follow that G is planar.

Proposition 12.26. *Every simple planar graph has a vertex of degree 5 or less.*

Proof. Let G be a simple planar graph. If $|V(G)| \leq 2$, then $d_G(v) \leq 2$ for all $v \in V(G)$ since G is simple, and there is nothing left to prove. We henceforth assume that $|V(G)| \geq 3$, and assume further that G is connected. Suppose $d_G(v) \geq 6$ for all $v \in V(G)$. By Theorem 10.21,

$$2|E(G)| = \sum_{v \in V(G)} d_G(V) \geq \sum_{v \in V(G)} 6 = 6|V(G)|,$$

so that $|E(G)| \geq 3|V(G|$, and hence $|E(G)| > 3|V(G)| - 6$. But this contradicts Proposition 12.23, which compels us to conclude there exists some $u \in V(G)$ such that $d_G(u) \leq 5$ in the case when G is connected.

Finally, assume that G is disconnected, and let H be a connected component. Either $|V(H)| \leq 2$ or $|V(H)| \geq 3$, but whatever the case may be, by our previous arguments

[16]The reason for this nomenclature will be revealed in the next section.

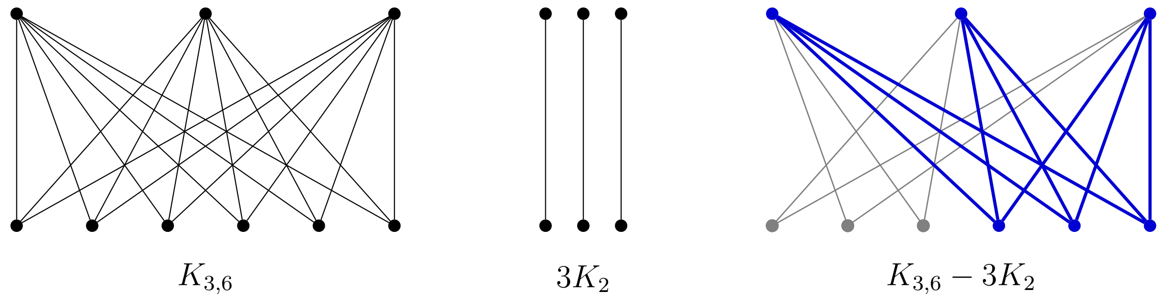


FIGURE 30. $K_{3,3}$, depicted with bold edges at right, is a subgraph of $K_{3,6} - 3K_2$,

there exists some $u \in V(H)$ such that $d_H(u) \leq 5$, and the proof is done by observing that $d_G(u) = d_H(u)$. ■

Example 12.27. Show that if G is a simple connected graph with $|V(G)| = 8$ and $|E(G)| \geq 19$, then G is nonplanar.

Solution. We have $|E(G)| \geq 19 > 18 = 3|V(G)| - 6$, so G is nonplanar by Proposition 12.23, and the proof is done.

It is a fact that there are 838 distinct simple connected graphs that have 8 vertices and at least 19 edges. We have thus shown that all of them are nonplanar in a single swoop. ■

Example 12.28. Let $3K_2$ be the graph that is the disjoint union of three copies of K_2. Given graph G with subgraph H, the graph $G - E(H)$ may be written as $G - H$. See Definition 10.37. Now, $K_{3,6}$ certainly has a subgraph that is isomorphic to $3K_2$, so that we may meaningfully consider the graph $K_{3,6} - 3K_2$ and ask whether or not it is planar. If $K_{3,6}$ has bipartition (U, V) with $U = \{u_1, u_2, u_3\}$ and $V = \{v_1, \ldots, v_6\}$, then $K_{3,6} - 3K_2$ may be characterized as the graph $K_{3,6} - S$, where $S = \{u_1v_1, u_2v_2, u_3v_3\}$. If $S' = \{u_1v_4, u_2v_5, u_3v_6\}$, then $K_{3,6} - S'$ is isomorphic to $K_{3,6} - S$ and thus is another instantiation of $K_{3,6} - 3K_2$. Indeed, $K_{3,6} - 3K_2$, like $K_{3,6}$ and K_2 themselves, may be regarded as an isomorphism class of graphs.

Now to the question of whether $K_{3,6} - 3K_2$ is planar or not. It will be convenient for vertices to have labels, so we shall take $K_{3,6}$ to have the aforementioned bipartition (U, V), let S be as before, and examine $K_{3,6} - S$. Among the edges in the graph we have u_1v_k, u_2v_k, u_3v_k for $k \in \{4, 5, 6\}$, and these 9 edges and their endpoints together constitute the graph $K_{3,3}$. Thus $K_{3,3}$ is a subgraph of $K_{3,6} - 3K_2$, and since $K_{3,3}$ is nonplanar by Proposition 12.25, it follows by Proposition 12.16 that $K_{3,6} - 3K_2$ is nonplanar. See Figure 30. ■

The **average degree** of a graph G, denoted by $\bar{d}(G)$, is simply the average of the degrees of the vertices of G; that is,

$$\bar{d}(G) = \frac{1}{|V(G)|} \sum_{v \in V(G)} d_G(v).$$

The following theorem furnishes an upper bound on $\bar{d}(G)$, and also immediately implies Proposition 12.26.

Theorem 12.29. *If G is a connected simple planar graph, then $\bar{d}(G) < 6$.*

Proof. If $|V(G)| \leq 2$, then $d_G(v) \leq 2$ for all $v \in V(G)$ since G is simple, and therefore $\bar{d}(G) < 6$. Suppose $G = (V, E)$ is a connected simple planar graph with at least three vertices. Using Theorem 10.21 followed by Proposition 12.23, we have

$$\bar{d}(G) = \frac{1}{|V|} \sum_{v \in V} d_G(v) = \frac{2|E|}{|V|} \leq \frac{2(3|V| - 6)}{|V|} = 6 - \frac{12}{|V|} < 6,$$

as desired. ■

12.3 – Subdivisions and Homeomorphisms

In this section we state Kuratowski's theorem, which is a momentous result in the theory of planar graphs. In the literature the theorem is typically expressed using the concept of graph homeomorphism, which in turn depends on the notion of an edge subdivision. Before we present the theorem, therefore, we need to establish some definitions. We will also establish a few propositions, one of which will be needed to prove part of Kuratowski's theorem.

Definition 12.30. *Let G be a graph with incidence function ι, and let $e \in E(G)$ be such that $\iota(e) = \{u, v\}$ (with $u = v$ allowed). To* ***subdivide the edge*** *e means to remove e from $E(G)$, add a new vertex w to $V(G)$, then add distinct edges e_1 and e_2 to $E(G)$ such that $\iota(e_1) = \{u, w\}$ and $\iota(e_2) = \{w, v\}$.*

Another term for subdividing an edge is **edge-subdivision**. To **subdivide a graph** G means to perform a finite sequence of edge-subdivision operations on G, with the resulting graph being called a **subdivision** of G. Since the empty sequence (i.e. the sequence containing no terms) is considered a finite sequence, *we always consider G itself to be a subdivision of G.*

Something like the inverse of the edge-subdivision operation is the "vertex smooth out" (or "vertex smooth away") operation.

Definition 12.31. *Let G be a graph with incidence function ι, with $w \in V(G)$ such that $d_G(w) = 2$, and $e_1, e_2 \in E(G)$ such that $\iota(e_1) = \{u, w\}$ and $\iota(e_2) = \{w, v\}$. To* ***smooth out the vertex*** *w means to remove w from $V(G)$ and both e_1 and e_2 from $E(G)$, then add a new edge e to $E(G)$ such that $\iota(e) = \{u, v\}$.*

Another term for smoothing out a vertex is **vertex-smoothing**. To **smooth out a graph** G means to perform a finite sequence of vertex-smoothing operations on G, with the resulting graph being called a **smoothing** of G.

Any finite sequence of edge-subdivision and vertex-smoothing operations performed on a planar graph G results in a new graph G' that is likewise planar. To show this, it is enough to show that any single edge-subdivision or vertex-smoothing operation preserves the planarity of G.

Let G have incidence function ι, let $\mathfrak{e}$ be any embedding of G in the plane, and for $e \in E(G)$ let $\iota(e) = \{u, v\}$. The embedding $\mathfrak{e}$ maps $u, v \in V(G)$ as points $p_u, p_v \in \mathbb{R}^2$, and maps e as a curve γ_e in $\mathbb{R}^2$ from p_u to p_v (with $p_u = p_v$ if and only if $u = v$). The curve γ_e may be parametrized by a vector-valued function $\gamma_e : [0,1] \to \mathbb{R}^2$ of the form $\gamma_e(t) = (x(t), y(t))$ for $0 \le t \le 1$, with $\gamma_e|_{(0,1)}$ being one-to-one so that the curve does not cross itself. (It is often desirable, though not always essential, that the parametrization have $[0,1]$ as its domain.) Specifically we have $\mathfrak{e}(u) = p_u$, $\mathfrak{e}(v) = p_v$, and $\mathfrak{e}(e) = \gamma_e$, where $\gamma_e(0) = p_u$ and $\gamma_e(1) = p_v$. We now subdivide e once, precisely in the manner described in Definition 12.30, in order to obtain a new graph G'. We now find an embedding $\mathfrak{e}' : G' \to \mathbb{R}^2$ of G' so as to demonstrate the planarity of G'. Indeed, much is the same: $\mathfrak{e}'(v') = \mathfrak{e}(v')$ for all $v' \in V(G') - \{w\} = V(G)$, and $\mathfrak{e}'(e') = \mathfrak{e}(e')$ for all $e' \in E(G') - \{e, e_1, e_2\}$. But how to define $\mathfrak{e}'(w)$, $\mathfrak{e}'(e_1)$, and $\mathfrak{e}'(e_2)$? We could do this: $\mathfrak{e}'(w) = \gamma_e(s)$ for some $0 < s < 1$, $\mathfrak{e}'(e_1) = \gamma_e|_{[0,s]}$, and $\mathfrak{e}'(e_2) = \gamma_e|_{[s,1]}$. If γ_e

happens to be parametrized by

$$\gamma_e(t) = (1-t)p_u + tp_v, \quad t \in [0,1],$$

which is a line segment from p_u to p_v, then letting $\mathfrak{e}'(w) = \gamma_e(1/2)$ places the image of the new vertex w in the drawing $\mathfrak{e}'(G')$ right where the midpoint of the image of the edge e in the drawing $\mathfrak{e}(G)$ is located. In any case the point $p_w := \mathfrak{e}'(w)$ will be located in $\mathfrak{e}'(G')$ at coordinates that the curve γ_e passes through in $\mathfrak{e}(G)$, *excluding the endpoints.* Meanwhile $\mathfrak{e}'(e_1) = \gamma_e|_{[0,s]}$ traces the old curve $\gamma_e : [0,1] \to \mathbb{R}^2$ from p_u to p_w, and $\mathfrak{e}'(e_2) = \gamma_e|_{[s,1]}$ traces γ_e from p_w to p_v. (We may reparametrize the new curves $\gamma_e|_{[0,s]}$ and $\gamma_e|_{[s,1]}$ so that their domains are both $[0,1]$, if desired.) Since the new edges e_1 and e_2 in G' have images that follow the footsteps of the image of the deleted edge e, with the curve for e_1 ending at the same point p_w where the curve for e_2 begins, the replacing of e with w, e_1 and e_2 introduces no edge-crossings as we pass from the planar drawing $\mathfrak{e}(G)$ to the new drawing $\mathfrak{e}'(G')$. Therefore $\mathfrak{e}'(G')$ is a planar drawing of G' (i.e. $\mathfrak{e}'$ is an embedding of G' in the plane), and we conclude that the subdivision G' is planar.

Now we suppose that we've smoothed out a vertex w in a planar graph G, as described in Definition 12.31. Let $\mathfrak{e}$ be an embedding of G, with $p_u = \mathfrak{e}(u)$, $p_v = \mathfrak{e}(v)$, $p_w = \mathfrak{e}(w)$, $\gamma_{e_1} = \mathfrak{e}(e_1)$, and $\gamma_{e_2} = \mathfrak{e}(e_2)$. We define $\gamma_{e_1}, \gamma_{e_2} : [0,1] \to \mathbb{R}^2$ such that $\gamma_{e_1}(0) = p_u$, $\gamma_{e_1}(1) = \gamma_{e_2}(0) = p_w$, and $\gamma_{e_2}(1) = p_v$. We now define an embedding $\mathfrak{e}'$ of the graph G' that results from smoothing out w as follows: $\mathfrak{e}'(v') = \mathfrak{e}(v')$ for all $v' \in V(G')$, $\mathfrak{e}'(e') = \mathfrak{e}(e')$ for all $e' \in E(G') - \{e\}$, and $\mathfrak{e}'(e) = \gamma_e$ for $\gamma_e : [0,1] \to \mathbb{R}^2$ given by

$$\gamma_e(t) = \begin{cases} \gamma_{e_1}(2t), & t \in [0,1/2) \\ \gamma_{e_2}(2t-1), & t \in [1/2,1]. \end{cases}$$

The curve γ_e passes through the same points in $\mathbb{R}^2$ as the two curves γ_{e_1} and γ_{e_2} taken together, so that $\gamma_e([0,1]) = \gamma_{e_1}([0,1]) \cup \gamma_{e_2}([0,1])$. Thus no edge-crossings are introduced in the drawing $\mathfrak{e}'(G')$, so that $\mathfrak{e}'$ is indeed an embedding of G', and therefore G' is planar.

It is evident that if G is subdivided as in Definition 12.30 to obtain G', then smoothing out w in G' will restore G.

We now have established that a single instance of subdividing or smoothing a planar graph G results in a new graph G' that is also planar. With this fact in hand it is straightforward to prove the following proposition by induction on the total number of edge-subdivision and vertex-smoothing operations applied to G.

Proposition 12.32. *A finite sequence of edge-subdivision and vertex-smoothing operations performed on a planar graph G results in a new graph that is also planar.*

Proposition 12.33. *A subdivision of a graph G is planar if and only if G itself is planar.*

Proof. Suppose a subdivision G' of the graph G is planar. Since G may be obtained from G' by performing a finite sequence of vertex-smoothing operations, Proposition 12.32 implies that G is planar. Conversely, if G is planar, then certainly any subdivision G' is planar by the same proposition. ∎

Definition 12.34. *Graphs G and H are **homeomorphic** if a subdivision of G is isomorphic to a subdivision of H.*

Proposition 12.35. *Any graph is homeomorphic to any of its subdivisions.*

Proof. Let G be a graph, and let G' be a subdivision of G. We must show that a subdivision of G is isomorphic to a subdivision of G'. This can be accomplished by showing that G' is isomorphic to a subdivision of G'. Indeed, G' is considered to be a subdivision of itself, and so, since it is clear that G' is isomorphic to itself, the proof is done. ■

As one might surmise, since a graph isomorphism preserves edges, any graph that is isomorphic to a planar graph is itself planar. This fact is needed to prove the following.

Proposition 12.36. *Let G and H be homeomorphic graphs. Then G is planar if and only if H is planar.*

Proof. Suppose that G is planar. Since G and H are homeomorphic, some subdivision G' of G is isomorphic to some subdivision H' of H. By Proposition 12.33, G' is planar, and hence H' is planar since $G' \simeq H'$. Therefore H is planar by Proposition 12.33. The proof of the converse is much the same. ■

Proposition 12.37. *If a graph G contains a subgraph that is homeomorphic to a nonplanar graph, then G is nonplanar.*

Proof. Suppose graph G contains a subgraph H that is homeomorphic to a nonplanar graph. Then H is nonplanar by Proposition 12.36, and therefore G is likewise nonplanar by Proposition 12.16. ■

Corollary 12.38. *If a graph G contains a subgraph that is homeomorphic to K_5 or $K_{3,3}$, then G is nonplanar.*

Proof. Suppose G contains a subgraph that is homeomorphic to K_5 or $K_{3,3}$. Both of these graphs are nonplanar by Proposition 12.25, and thus G itself is nonplanar by Proposition 12.37. ■

The contrapositive of Corollary 12.38 states that if G is planar, then G contains no subgraph that is homeomorphic to K_5 or $K_{3,3}$. It turns out that the converse of this statement is true as well, but it is considerably more difficult to prove. Kasimir Kuratowski was the first to publish a proof in 1930, and we present now his celebrated theorem.

Theorem 12.39 (Kuratowski's Theorem). *A graph is planar if and only if it contains no subgraph homeomorphic to K_5 or $K_{3,3}$.*

Definition 12.40. *The* ***barycentric subdivision*** *of a graph G is the subdivision of G obtained by subdividing every edge of G.*

When the operation of barycentric subdivision is executed once on a graph G, sometimes the graph G' that results is called the **first barycentric subdivision** of G. If barycentric subdivision is then carried out on G', the resultant graph G'' is then called the **second barycentric subdivision** of G, and so on. Of great utility is barycentric subdivision's effect of transforming a general graph featuring loops or parallel edges into a simple graph. Indeed, the first barycentric subdivision of any general graph G will always be loopless, and the second barycentric subdivision will always be a simple graph. We will prove these assertions presently, but first we will establish another fact, namely that the barycentric subdivision of any graph is bipartite.

Proposition 12.41. *The barycentric subdivision of any graph is bipartite.*

Proof. Let G be a graph, and let G' be the barycentric subdivision of G. Define $V = V(G)$ and $W = V(G') - V$, so that W is the set of the new vertices inserted into G to effect the subdivision. Now, any $e' \in E(G')$ derives from the edge-subdivision of some $e \in E(G)$, so that e' has one endpoint in V and the other in W. Therefore G' is bipartite, with bipartition (V, W). ■

Proposition 12.42. *The barycentric subdivision of any graph is loopless.*

Proof. Let G be a graph, and let ℓ be a loop in G at some vertex v. Applying the operation of edge-subdivision on ℓ, a new vertex w is introduced, whereafter the loop ℓ is replaced by two parallel edges in G' having endpoints v and w. Therefore G' is loopless. ■

Proposition 12.43. *The barycentric subdivision of any loopless graph is a simple graph.*

Proof. Let G be a loopless graph, and let G' be the barycentric subdivision of G. No edge-subdivision operation can possibly result in parallel edges where none existed before, and so if G happens to be a simple graph, by Proposition 12.42 it follows that G' is simple.

Now suppose that $e_1, e_2 \in E(G)$ are parallel edges having $u, v \in V(G)$ as endpoints, where $u \neq v$ since G is loopless. Barycentric subdivision inserts two new vertices w_1 and w_2 into G, then replaces e_1 with $e_1' = uw_1$ and $e_1'' = vw_1$, and e_2 with $e_2' = uw_2$ and $e_2'' = vw_2$. Since the vertices u, v, w_1, and w_2 are distinct, none of the four new edges is parallel to another, and therefore G' can have no parallel edges. Again, Proposition 12.42 guarantees that G' also has no loops, and therefore G' is simple. ■

In the course of proving the next proposition, we make use of the fact that, for $n \geq 2$, the nth barycentric subdivision of a graph G is the first barycentric subdivision of the $(n-1)$st barycentric subdivision of G.

Proposition 12.44. *The second barycentric subdivision of any graph is a simple graph.*

Proof. Let G be a graph, and let G' and G'' be the first and second barycentric subdivision of G, respectively. By Proposition 12.42, G' is loopless; and since G'' is the first barycentric subdivision of G', Proposition 12.43 implies that G'' is a simple graph. ■

Proposition 12.45. *Every graph is homeomorphic to a bipartite graph.*

Proof. Let G be a graph. By Proposition 12.41 the barycentric subdivision of G is a bipartite graph, and since G is homeomorphic to any subdivision of itself by Proposition 12.35, the proof is done. ■

12.4 – Vertex Colorings

Recall that when partitioning a nonempty set S, the mutually disjoint sets constituting the partition are called **cells**. Given a graph G, what is the minimum number of cells that a partition of the vertex set $V(G)$ must have in order for it to be possible that no edge in $E(G)$ has both endpoints located at vertices in the same cell? If we assign a different color to the vertices in each cell, so that all vertices in one cell are red, all vertices in another cell are blue, and so on, then all the vertices in G will be given a color since the union of all the cells must equal $V(G)$. Coloring all the vertices of a graph is therefore equivalent to a partitioning of that graph's vertex set, with the use of colors merely serving as an aid in representing a partition in a more intuitive or visually appealing way.

Definition 12.46. *Let G be a graph and C a set of k distinct colors. A surjective function $\mathfrak{c} : V(G) \to C$ is a **vertex k-coloring** of G. The **color class** of a color $i \in C$ is the set V_i of all vertices assigned the color i, so that $V_i = \{v \in V(G) : \mathfrak{c}(v) = i\}$.*

*A **vertex coloring** of G is a vertex k-coloring of G for any $k \geq 1$. A **proper vertex coloring** of G is a vertex coloring such that no adjacent vertices are assigned the same color. If G has a proper vertex k-coloring, then it is said to be **vertex k-colorable**.*

Since vertex colorings of graphs is the only sort of coloring we entertain in this section, we shall often omit the word "vertex" when using many of the terms defined in this section. For instance, whenever we make mention of a "proper coloring," it is to be understood that we mean a proper *vertex* coloring. More care in this matter will be taken in the next section when our attention turns to map colorings.

Though the elements of the set C in Definition 12.46 are called colors, in practice these elements are typically positive integers. Thus if $C = \{1, 2, \ldots, k\}$, then C contains the "colors" $1, 2, \ldots, k$. It would, of course, be impractical to employ literal colors in the case when, say, $k = 100$.

If a graph G happens to have a loop at $v \in V(G)$, then the graph is "uncolorable" in the sense that it can have no proper coloring: v is, after all, adjacent to itself.

For the statement of the next proposition we make the following definition: A set of vertices S in a graph G is **independent** (or **stable**) if no two distinct vertices in S are joined by an edge. The **independence number** (or **stability number**) of G, denoted by $\alpha(G)$, is the number of vertices in an independent set in G of maximum cardinality.

Proposition 12.47. *Let $\mathfrak{c}$ be a vertex k-coloring of graph G. Then $\mathfrak{c}$ is a proper vertex k-coloring of G if and only if each of the color classes associated with $\mathfrak{c}$ is an independent set of vertices.*

Proof. Suppose $\mathfrak{c}$ is a proper k-coloring of G. Let V_i be a color class associated with $\mathfrak{c}$, and suppose $u, v \in V_i$. Thus u and v are both assigned the same color i, and there can be no edge in G joining u and v since $\mathfrak{c}$ is a proper coloring. Therefore V_i is an independent set of vertices.

For the converse, suppose $\mathfrak{c}$ is not a proper k-coloring of G. Then there exist vertices $u, v \in V(G)$ that have been assigned the same color, such as i, and yet are joined by an edge. Since both u and v are in the same color class V_i, it follows that V_i is not an independent set of vertices. ■

Definition 12.48. *The **vertex chromatic number** of a graph G, denoted by $\chi(G)$, is the minimum number of distinct colors required for a proper vertex coloring of G. If $\chi(G) = k$, then G is **vertex k-chromatic**. A **chromatic coloring** of G is a proper coloring of G that uses precisely $\chi(G)$ colors.*

We observe that if graph G is k-chromatic, so that $\chi(G) = k$, then G has no proper ℓ-coloring for any $\ell < k$.

To determine $\chi(G)$ for a given graph G is often no trivial matter. A common approach is to first obtain an upper bound for $\chi(G)$ by finding a proper k-coloring of G, so that $\chi(G) \leq k$; then, a subgraph H of G is identified for which it may be readily demonstrated that $\chi(H) = k$, whereupon the following proposition is used.

Proposition 12.49. *If H is a subgraph of graph G, then $\chi(G) \geq \chi(H)$.*

Proof. Suppose H is a subgraph of G. Suppose $\chi(G) = k < \chi(H)$, so that the minimum number of colors required to achieve a proper coloring of G is k. For $C = \{1, 2, \ldots, k\}$, let $\mathfrak{c} : V(G) \to C$ be a proper k-coloring of G, and let $C' = \mathfrak{c}(V(H)) \cap C$. Then $\mathfrak{c}|_{V(H)} : V(H) \to C'$ is a proper $|C'|$-coloring of H for $|C'| \leq k$, and hence $\chi(H) \leq k$. Having arrived at a contradiction, we conclude that $\chi(G) \geq \chi(H)$. ■

We give now a few more propositions that impose upper or lower bounds on $\chi(G)$. As in §11.1 the symbol $\Delta(G)$ denotes the largest degree attained by the vertices of an undirected graph G; that is, $\Delta(G) = \max\{d_G(v) : v \in V(G)\}$.

Proposition 12.50. *If G is a simple graph, then $\chi(G) \leq \Delta(G) + 1$.*

Proof. Suppose G is a simple graph, with $V(G) = \{1, 2, \ldots, n\}$. Let C be a set of the form $\{1, 2, \ldots, k\}$. We construct a vertex coloring $\mathfrak{c} : V(G) \to C$ of G as follows: let $\mathfrak{c}(v_1) = 1$, and for each $2 \leq i \leq n$, starting at 2 and proceeding in ascending order to n, let $\mathfrak{c}(v_i)$ equal the smallest color number in C that has not been assigned to any vertex in $S_i := \{v_1, \ldots, v_{i-1}\} \cap N_G(v_i)$. This "sequential vertex coloring" procedure results in a proper coloring of G. Now, $|S_i| \leq \Delta(G)$ since $|N_G(v_i)| \leq \Delta(G)$, and so at worst the colors $1, \ldots, \Delta(G)$ have been assigned to the vertices in S_i, so that v_i, being adjacent to them all, is assigned the color numbered $\Delta(G) + 1$. ■

Proposition 12.51. *If G is a graph with k mutually adjacent vertices, then $\chi(G) \geq k$.*

Proof. Suppose G contains a set of vertices $V = \{v_1, \ldots, v_k\}$ that are mutually adjacent, so that each v_i is joined to all vertices in $V - \{v_i\}$. If fewer than k colors are assigned to the vertices in $V(G)$, then at least two vertices $u, v \in V$ must be the same color. But u and v are adjacent, so that the resultant coloring of G cannot be proper. Therefore any proper coloring of G must employ at least k colors, so that $\chi(G) \geq k$. ■

Example 12.52. In Figure 31 is shown a proper vertex 4-coloring of a graph G using the color set $C = \{1, 2, 3, 4\}$. The four shaded vertices are mutually adjacent, and so $\chi(G) \geq 4$ by Proposition 12.51. On the other hand, the fact that there exists a proper vertex 4-coloring of G implies that $\chi(G) \leq 4$. Therefore it must be that $\chi(G) = 4$. ■

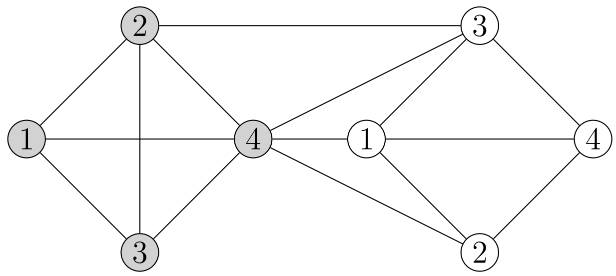


Figure 31.

A **clique** in a graph G is a maximal set of mutually adjacent vertices in G. Thus $S \subseteq V(G)$ is a clique in G if S is a set of mutually adjacent vertices in G, and there exists no $S' \subseteq V(G)$ that is a set of mutually adjacent vertices such that $S' \supset S$. The **clique number** of G, denoted by $\omega(G)$, is the number of vertices in a clique in G of maximum cardinality.

Corollary 12.53. *If G is a graph, then $\chi(G) \geq \omega(G)$.*

Proof. The corollary follows as a direct consequence of Proposition 12.51 and the definition of clique number. ■

Proposition 12.54. *If G is a graph, then*

$$\chi(G) \geq \left\lceil \frac{|V(G)|}{\alpha(G)} \right\rceil. \tag{12.4}$$

Proof. Let $\mathfrak{c} : V(G) \to C$ be a proper coloring of G. For each $i \in C$, the color class V_i defined by $\mathfrak{c}$ is an independent set by Proposition 12.47, so that $|V_i| \leq \alpha(G)$, which in turn implies that

$$\sum_{i \in C} |V_i| \leq \sum_{i \in C} \alpha(G) \quad \hookrightarrow \quad |V(G)| \leq \alpha(G)|C| \quad \hookrightarrow \quad |C| \geq \frac{|V(G)|}{\alpha(G)},$$

and finally

$$|C| \geq \left\lceil \frac{|V(G)|}{\alpha(G)} \right\rceil$$

since $|C|$ must be an integer. Since $\mathfrak{c}$ is an arbitrary proper coloring, the inequality (12.4) follows. ■

Proposition 12.55. *If G is a graph, then $\chi(G) = 1$ if and only if $E(G) = \varnothing$.*

Proof. Suppose $\chi(G) = 1$. Then G has no loops, since otherwise G would be uncolorable and hence $\chi(G)$ undefined. Also G has no edges that are not loops, since otherwise two distinct vertices would be joined by an edge and so need to be assigned two different colors in a proper coloring, implying $\chi(G) \geq 2$. Therefore $E(G) = \varnothing$.

Now suppose that $E(G) = \varnothing$. Then no vertex is joined to any other, and so may all be assigned the same color in a proper coloring. Therefore $\chi(G) = 1$. ■

The next proposition is in essence Theorem 10.12, one of the first results we stated and proved in our study of graph theory.

Proposition 12.56. *Suppose G is a graph with $E(G) \neq \varnothing$. Then G is bipartite if and only if $\chi(G) = 2$.*

Proof. Suppose G is bipartite, and let (V_1, V_2) be a bipartition of $V(G)$. Since a bipartite graph cannot have loops, we have $\chi(G) \geq 2$ by Proposition 12.55. Now, both V_1 and V_2 are nonempty since $E(G) \neq \varnothing$, and so a proper 2-coloring may be realized by assigning one color to all the vertices in V_1, and another color to all the vertices in V_2. Therefore $\chi(G) = 2$.

For the converse, suppose $\chi(G) = 2$. Let V_1 and V_2 be the two color classes associated with a chromatic coloring of G. Then (V_1, V_2) is a bipartition of $V(G)$, and therefore G is bipartite. ■

A graph G is a **path graph** if there exists a path that contains all the vertices and edges of G. We let P_n denote a path graph comprised of n vertices and $n - 1$ edges.

Corollary 12.57. *Let G be a graph with at least one edge.*

1. *If G is a path graph P_n for $n \geq 2$, then $\chi(P_n) = 2$.*
2. *If G is a tree T, then $\chi(T) = 2$.*
3. *If G is a hypercube graph Q_n, then $\chi(Q_n) = 2$.*
4. *If G is an even cycle C_{2n} for $n \geq 1$, then $\chi(C_{2n}) = 2$.*

Proof.
Proof of (1). Let P_n be a path graph for $n \geq 2$, so that $E(P_n) \geq 1$, and P_n has vertex-edge sequence $v_1 e_1 v_2 \cdots v_{n-1} e_{n-1} v_n$. Defining $V = \{v_k : k \text{ is odd}\}$ and $W = \{v_k : k \text{ is even}\}$, we find (V, W) to be a bipartition of $V(P_n) = \{v_1, \ldots, v_n\}$, and hence P_n is a bipartite graph. Therefore $\chi(P_n) = 2$ by Proposition 12.56.

Proof of (2). Suppose T is a tree. By definition T is acyclic, hence has no odd cycles, and so is bipartite by Theorem 10.42. Therefore $\chi(T) = 2$ by Proposition 12.56.

Proof of (3). Any hypercube graph Q_n is bipartite by Proposition 10.15, and therefore $\chi(Q_n) = 2$ by Proposition 12.56.

Proof of (4). Suppose C_{2n} is an even cycle with $2n$ vertices. These vertices may be labeled $v_1, \ldots, v_{2n}$ such that $E(C_{2n}) = \{v_i v_{i+1} : 1 \leq i \leq 2n\}$, where we define $v_{2n+1} = v_1$. If we then define $V_1 = \{v_{2i} : 1 \leq i \leq n\}$ and $V_2 = \{v_{2i+1} : 0 \leq i \leq n\}$, then V_1 and V_2 are independent sets such that (V_1, V_2) forms a bipartition of $V(G)$. Therefore C_{2n} is a bipartite graph, and $\chi(C_{2n}) = 2$ by Proposition 12.56. ■

Proposition 12.58. *If C_{2n+1} is an odd cycle for $n \geq 1$, then $\chi(C_{2n+1}) = 3$.*

Proof. The vertices of C_{2n+1} may be labeled $v_1, \ldots, v_{2n+1}$ such that $C_{2n+1} = v_1 v_2 \cdots v_{2n+1} v_1$. Two independent sets in C_{2n+1} are $V_1 = \{v_{2i} : 1 \leq i \leq n\}$ and $V_2 = \{v_{2i-1} : 1 \leq i \leq n\}$. Both sets have cardinality n, and since there is no independent set of greater cardinality we have $\alpha(C_{2n+1}) = n$. Now, by Proposition 12.54,

$$\chi(C_{2n+1}) \geq \left\lceil \frac{|V(C_{2n+1})|}{\alpha(C_{2n+1})} \right\rceil = \left\lceil \frac{2n+1}{n} \right\rceil = 3;$$

on the other hand, $\chi(C_{2n+1}) \leq \Delta(C_{2n+1}) + 1 = 3$ by Proposition 12.50. Therefore it must be that $\chi(C_{2n+1}) = 3$. ■

The **join** of graphs G and H, denoted by $G+H$, is constructed from the disjoint union $G \sqcup H$ of the graphs by joining each vertex in G to each vertex in H. Thus $V(G+H) = V(G) \sqcup V(H)$ and

$$E(G+H) = E(G) \sqcup E(H) \sqcup \{uv : u \in V(G) \text{ and } v \in V(H)\}.$$

If v is a vertex not in graph G, then the **join** of v to G, denoted by $G+v$, is constructed from G by joining v to each vertex in G. Thus $V(G+v) = V(G) \cup \{v\}$ and

$$E(G+v) = E(G) \sqcup \{uv : u \in V(G)\}.$$

It can be seen that $G + v \simeq G + K_1$ in any case.

Lemma 12.59. *Each color class associated with a proper vertex coloring of $G+H$ is a subset of either $V(G)$ or $V(H)$.*

Proof. Let $V \subseteq V(G+H)$ be a color class associated with a proper coloring of $G+H$. Suppose $V \cap V(G) \neq \varnothing$ and $V \cap V(H) \neq \varnothing$, so there exists some $u \in V(G)$ in V, and also some $v \in V(H)$ in V. But in $G+H$ the vertices u and v are joined by an edge, and since these vertices have been assigned the same color, the vertex coloring of $G+H$ is not proper. This being a contradiction, either $V \cap V(G)$ or $V \cap V(H)$ must be empty, and therefore either $V \subseteq V(G)$ or $V \subseteq V(H)$. ■

Proposition 12.60. *For any graphs G and H, $\chi(G+H) = \chi(G) + \chi(H)$.*

Proof. Since G and H are both subgraphs of $G+H$, any proper coloring of $G+H$ must necessarily define a proper coloring of both G and H. However, when constructing a proper coloring of $G+H$, Lemma 12.59 makes clear that none of the colors used for a proper coloring of G can be the same as any of the colors used for a proper coloring of H. Thus any proper coloring of $G+H$ must utilize at least $\chi(G)+\chi(H)$ colors, so that $\chi(G+H) \geq \chi(G)+\chi(H)$.

Next, define a chromatic coloring $\mathfrak{c}_G : V(G) \to C_G$ of G, and also a chromatic coloring $\mathfrak{c}_H : V(H) \to C_H$ of H, such that $C_G \cap C_H = \varnothing$. Since $|C_G| = \chi(G)$ and $|C_H| = \chi(H)$, the total number of colors used is $\chi(G)+\chi(H)$. Adding edges to $G \sqcup H$ to produce $G+H$, we find that the colors assigned to the vertices of $G+H$ by $\mathfrak{c}_G$ and $\mathfrak{c}_H$ define a proper coloring of $G+H$. Thus a chromatic coloring of $G+H$ must use at most $\chi(G)+\chi(H)$ colors, implying that $\chi(G+H) \leq \chi(G)+\chi(H)$. ■

Proposition 12.61. *If W_{2n} is an even wheel for $n \geq 1$, then $\chi(W_{2n}) = 3$.*

Proof. We observe that $W_{2n} = C_{2n} + K_1$. By Corollary 12.57, $\chi(C_{2n}) = 2$; and by Proposition 12.55, $\chi(K_1) = 1$. Therefore $\chi(W_{2n}) = \chi(C_{2n}+K_1) = \chi(C_{2n}) + \chi(K_1) = 3$ by Proposition 12.60. ■

Proposition 12.62. *If W_{2n+1} is an odd wheel for $n \geq 1$, then $\chi(W_{2n+1}) = 3$.*

Proof. By Propositions 12.58 and 12.55, respectively, $\chi(C_{2n+1}) = 3$ and $\chi(K_1) = 1$. Therefore $\chi(W_{2n+1}) = \chi(C_{2n+1}+K_1) = \chi(C_{2n+1}) + \chi(K_1) = 4$ by Proposition 12.60. ■

Proposition 12.63. *For any $n \geq 1$, $\chi(K_n) = n$.*

Proof. Since $\Delta(K_n) = n - 1$, Proposition 12.50 implies that $\chi(K_n) \leq \Delta(K_n) + 1 = n$. On the other hand, K_n has n mutually adjacent vertices, so that $\chi(K_n) \geq n$ by Proposition 12.51. ■

In 1941, Leonard Brooks proved a result that is a significant improvement over the upper bound on $\chi(G)$ furnished by Proposition 12.50. We state his theorem here without proof.

Theorem 12.64 (Brooks' Theorem). *If G is a simple connected graph that is neither complete nor an odd cycle, then $\chi(G) \leq \Delta(G)$.*

We now consider a few results concerning the effect, if any, that removing a vertex or edge from a graph G has on the graph's chromatic number $\chi(G)$. To start, upper and lower bounds on the possible change in $\chi(G)$ when a vertex or edge is deleted are given by the following.

Proposition 12.65. *Let G be a graph.*

1. *For any $v \in V(G)$, $\chi(G) - 1 \leq \chi(G - v) \leq \chi(G)$.*
2. *For any $e \in E(G)$, $\chi(G) - 1 \leq \chi(G - e) \leq \chi(G)$.*

Proof. That $\chi(G - v) \leq \chi(G)$ and $\chi(G - e) \leq \chi(G)$ follows from Proposition 12.49. Now, for any $v \in V(G)$, the minimum number of colors required for a proper coloring of $G - v$ is either 1 less than the minimum required for G (so that $\chi(G - v) = \chi(G) - 1$), or else it is the same as the minimum required for G (so that $\chi(G - v) = \chi(G) > \chi(G) - 1$). Hence $\chi(G) - 1 \leq \chi(G - v)$, and by a similar argument $\chi(G) - 1 \leq \chi(G - e)$ for any $e \in E(G)$. ■

Since the vertex chromatic number of any graph G must be an integer whenever it exists, Proposition 12.65 implies that $\chi(G - v)$ must equal $\chi(G) - 1$ whenever it does not equal $\chi(G)$, and the same holds for $\chi(G - e)$.

Definition 12.66. *An edge e or vertex v in a graph G is **chromatically critical** if $\chi(G - e) = \chi(G) - 1$ or $\chi(G - v) = \chi(G) - 1$, respectively. We say G is **chromatically k-critical** if $\chi(G) = k$ and $\chi(H) < k$ for every proper subgraph H of G. A graph is **chromatically critical** if it is chromatically k-critical for some integer k.*

The word "chromatically" in the terms defined above is often omitted in practice, and some authors never include it. Our practice will be to omit it except in the statements of propositions, theorems, and the like.

Proposition 12.67. *For graph G let $e \in E(G)$. If e is a chromatically critical edge of G, then every chromatic coloring of $G - e$ must assign the same color to both endpoints of e.*

Proof. Suppose there is a chromatic coloring $\mathfrak{c} : V(G - e) \to C$ of $G - e$ that assigns different colors to the endpoints of e, so $|C| = \chi(G - e)$. Then $\mathfrak{c} : V(G) \to C$ is a proper coloring of G, so $\chi(G) \leq |C| = \chi(G - e)$, whereas $\chi(G - e) \leq \chi(G)$ by Proposition 12.49. Hence $\chi(G - e) = \chi(G) \neq \chi(G) - 1$, and we conclude that e is not a critical edge of G. ■

Proposition 12.68. *If G is a chromatically k-critical graph, then each vertex is chromatically critical.*

Proof. Suppose G is a chromatically k-critical graph, and fix $v \in V(G)$. Since $G - v$ is a proper subgraph of G we have $\chi(G - v) < k = \chi(G)$. Since $\chi(G - v)$ must equal either $\chi(G)$ or $\chi(G) - 1$ by Proposition 12.65, it follows that $\chi(G - v) = \chi(G) - 1$, and therefore v is chromatically critical. ■

Theorem 12.69. *If G is a chromatically k-critical graph, then $d_G(v) \geq k - 1$ for all $v \in V(G)$.*

Proof. Suppose G is a chromatically k-critical graph, so $\chi(G) = k$. Assume there exists some $v \in V(G)$ such that $d_G(v) < k - 1$. Since $\chi(G - v) = k - 1$ by Proposition 12.68, there is a proper $(k-1)$-coloring $\mathfrak{c} : V(G - v) \to C$ of $G - v$. Observing that $|C| = k - 1$ and $|N_G(v)| \leq k - 2$, there exists some color $i \in C$ that has not been assigned to any of the neighbors of v, and so we extend the domain of $\mathfrak{c}$ to $V(G)$ by designating $\mathfrak{c}(v) = i$. Now $\mathfrak{c} : V(G) \to C$ is a proper $(k-1)$-coloring of G, which implies that $\chi(G) \leq k - 1 = \chi(G) - 1$. This being a contradiction, we conclude that $d_G(v) \geq k - 1$ for all $v \in V(G)$. ■

12.5 – Map Colorings

Let G be a graph and S a surface, and suppose there exists an embedding of G on S. This is to say there is a drawing function $\mathfrak{e} : V(G) \cup E(G) \to \mathcal{P}(S)$ that produces a drawing of G on S, denoted by $\mathfrak{e}(G)$ as in §12.2, that features no edge-crossings. The **map of G on S** corresponding to $\mathfrak{e}$ is the drawing $\mathfrak{e}(G)$ together with its set of faces, which we denote by $F_{\mathfrak{e}}(G)$. Of course, each face $f \in F_{\mathfrak{e}}(G)$ is a subset of S, and it is always true that $\mathfrak{e}(G) \cup \left(\bigcup F_{\mathfrak{e}}(G)\right)$ is equal to S itself, so we must be clear that by a "map of G on S" we do not mean simply the union of the drawing $\mathfrak{e}(G)$ with its faces. Rather, there is the union of curves that is the drawing $\mathfrak{e}(G)$, as in Equation (12.2), and coupled with this drawing is the collection of regions in S that are the drawing's faces. In this sense the map of G on S that is associated with a particular drawing function $\mathfrak{e}$ is an ordered pair $(\mathfrak{e}(G), F_{\mathfrak{e}}(G))$, just as the graph G itself is an ordered pair $(V(G), E(G))$. For brevity we shall denote $(\mathfrak{e}(G), F_{\mathfrak{e}}(G))$ by the symbol $M_{\mathfrak{e}}(G)$.

Two faces of a map $M_{\mathfrak{e}}(G)$ are said to **meet** if their boundary walks share an edge in common. This is to say faces $f_1, f_2 \in F_{\mathfrak{e}}(G)$ meet if and only if there exists an edge $e \in E(G)$ such that the curve $\mathfrak{e}(e)$ that is the image of e on the surface S is a subset of $\partial f_1 \cap \partial f_2$. A single face of $M_{\mathfrak{e}}(G)$ is said to **meet itself** if its boundary walk traverses an edge more than once.

Definition 12.70. *Let $M_{\mathfrak{e}}(G)$ be a map of graph G on S, and let C be a set of k distinct colors. A surjective function $\mathfrak{c} : F_{\mathfrak{e}}(G) \to C$ is a **map k-coloring** of $M_{\mathfrak{e}}(G)$. The **color class** of a color $i \in C$ is the set F_i of all faces assigned the color i, so that $F_i = \{f \in F_{\mathfrak{e}}(G) : \mathfrak{c}(f) = i\}$.*

*A **map coloring** of G is a map k-coloring of $M_{\mathfrak{e}}(G)$ for any $k \geq 1$. A **proper map coloring** of $M_{\mathfrak{e}}(G)$ is a map coloring such that no faces in $F_{\mathfrak{e}}(G)$ that meet are assigned the same color. If $M_{\mathfrak{e}}(G)$ has a proper map k-coloring, then it is said to be **map k-colorable**.*

We emphasize that if a map $M_{\mathfrak{e}}(G)$ has a face that meets itself, then the map cannot be properly colored.

Definition 12.71. *The **chromatic number** of a map $M_{\mathfrak{e}}(G)$, denoted by $\chi(M_{\mathfrak{e}}(G))$, is the minimum number of distinct colors required for a proper map coloring of $M_{\mathfrak{e}}(G)$. If $\chi(M_{\mathfrak{e}}(G)) = k$, then $M_{\mathfrak{e}}(G)$ is **map k-chromatic**. A **chromatic coloring** of $M_{\mathfrak{e}}(G)$ is a proper coloring of $M_{\mathfrak{e}}(G)$ that uses precisely $\chi(M_{\mathfrak{e}}(G))$ colors.*

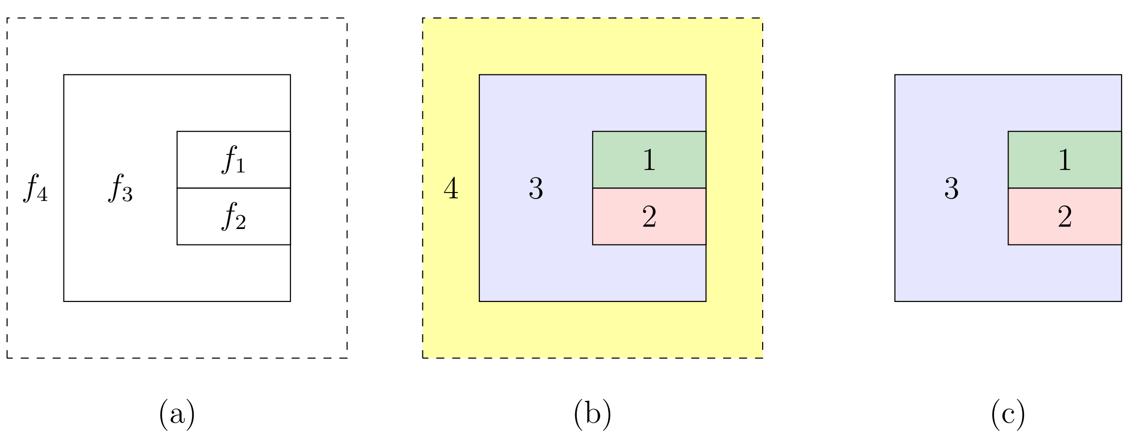


Figure 32.

A map on a plane may sometimes be considered without reference to a graph G, as in the next example, in which case we may simply denote the map by a symbol such as M.

Example 12.72. The chromatic number of the simple map M with face set $F = \{f_1, f_2, f_3, f_4\}$ in Figure 32(a) may be determined by inspection. Indeed, in Figure 32(b) is shown a proper map 4-coloring $\mathfrak{c} : F \to \{1, 2, 3, 4\}$ of M given by $\mathfrak{c}(f_i) = i$ for each $1 \leq i \leq 4$, and so $\chi(M) = 4$ since it is clear that no proper k-coloring of M exists for $k < 4$. However, if we exclude the unbounded external face f_4 of M from consideration (every map on a plane possesses precisely one external face), then what's left is a "bounded map" M' having only the faces f_1, f_2, and f_3, and a proper 3-coloring of M' is obtained simply by restricting $\mathfrak{c}$ to the face set $F - \{f_4\}$. See Figure 32(c). ■

It may not have escaped the reader's notice that many of the definitions we have furnished thus far in this section closely parallel definitions given in the previous section on vertex colorings. This may kindle a suspicion that some of the theory developed in §12.4 concerning vertex colorings may in similar fashion carry over to our study of map colorings. However, while we have established many results about graphs by this point, we have no results about maps. This observation suggests that the most efficient strategy should be the one that converts any problem in map coloring into an equivalent problem in vertex coloring. What is known as the **dual graph** of a map $M_{\mathfrak{e}}(G)$ makes this possible.

Remark 12.73. For the rest of this section all maps will be on the plane, so that the only surface S under consideration will be specifically $\mathbb{R}^2$.

We now describe what the dual graph of a map is. Let $M_{\mathfrak{e}}(G) = (\mathfrak{e}(G), F_{\mathfrak{e}}(G))$ be a map of G on $\mathbb{R}^2$, so that G is a planar graph with planar drawing $\mathfrak{e}(G)$. The dual graph of $M_{\mathfrak{e}}(G)$, denoted by G^*, is constructed as follows: First, for each $f \in F_{\mathfrak{e}}(G)$ let $f^* \in V(G^*)$, so that $V(G^*) = \{f^* : f \in F_{\mathfrak{e}}(G)\}$; then, for any $f_1, f_2 \in F_{\mathfrak{e}}(G)$, if the boundary walks of f_1 and f_2 have k edges in common, then $E(G^*)$ will contain precisely k edges with endpoints f_1^* and f_2^*. If a face f meets itself by virtue of its boundary walk traversing k edges more than once, then $E(G^*)$ will contain precisely k loops f^*f^*; that is, the dual graph G^* will have k distinct loops at the vertex f^*. The vertices, edges, and faces of the graph G are called **primal vertices**, **primal edges**, and **primal faces**, respectively, while the vertices, edges, and faces of the dual graph G^* are called **dual vertices**, **dual edges**, and **dual faces**, respectively.

From the description of how a dual graph is constructed, it is evident that if a map $M_{\mathfrak{e}}(G)$ has no face that meets itself, then the dual graph G^* will be a simple graph. Less obvious is that if $M_{\mathfrak{e}}(G)$ is a map of G on the plane, so that $\mathfrak{e}(G)$ is a planar drawing of G, then the dual graph G^* is necessarily also a planar graph. When constructing a dual graph of a map, it is usually desirable to present a drawing of the dual graph without edge-crossings, so that it is a planar drawing.

Example 12.74. In Figure 33(a) is shown a planar drawing $\mathfrak{e}(G)$ of a planar graph G, which together with the set of faces $F_{\mathfrak{e}}(G) = \{f_1, f_2, f_3, f_4, f_5\}$ constitutes a map of G on $\mathbb{R}^2$ that we denote by $M_{\mathfrak{e}}(G)$.

We now construct the dual graph of the map $M_{\mathfrak{e}}(G)$. In Figure 33(b) a dual vertex f_k^* is placed somewhere in the interior of each face $f_k \in F_{\mathfrak{e}}(G)$. Now, faces f_1 and f_2 meet since their

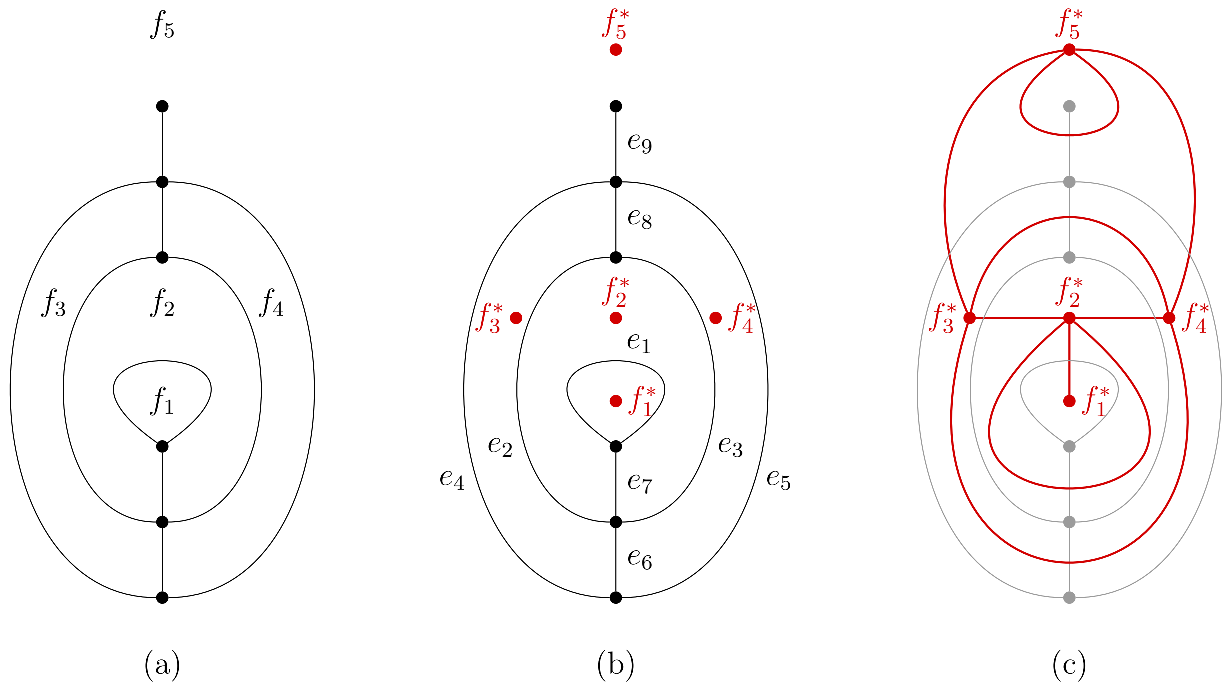


FIGURE 33. In (a) is shown a planar drawing of graph G. In (c) is shown, in red, a planar drawing of the dual graph G^*.

boundary walks have edge e_1 in common, and so the dual vertices f_1^* and f_2^* are joined by a single edge $f_1^* f_2^*$ in the dual graph G^*. In Figure 33(c) the image of this edge in depicted in red as an open curve that passes through the interior of the image of e_1. The endpoints of this curve are those points in the plane that are the images of vertices f_1^* and f_2^*, which we denote by these same symbols in the figure.

Faces f_2 meets faces f_3 and f_4 along edges e_2 and e_3, respectively, and so in Figure 33(c) we have curves representing the edges $f_2^* f_3^*$ and $f_2^* f_4^*$ in the planar drawing of G^*. Also f_2 meets itself along edge e_7, so there is a loop $f_2^* f_2^*$ in $E(G^*)$, and the image of this loop passes through the interior of e_7 while otherwise staying in the interior of f_2.

Faces f_3 and f_4 also meet, but in this case the meeting occurs along two edges: e_6 and e_8. Thus $E(G^*)$ will contain two distinct edges joining f_3^* and f_4^*. Correspondingly, there are two curves in the drawing of G^*: one passing through the interior of e_6, and the other passing through the interior of e_8. In Figure 33(c) these curves are drawn so that they otherwise remain in the interiors of f_3 and f_4. This is done to ensure that the resultant drawing of G^* has no edge-crossings and so is planar.

Finally, the exterior face f_5 meets itself along edge e_9, and meets faces f_3 and f_4 along e_4 and e_5, respectively. Thus $f_5^* f_5^*$, $f_3^* f_5^*$, and $f_4^* f_5^*$ are edges in $E(G^*)$, and Figure 33(c) depicts the images of these dual edges as curves passing through the interiors of the images of e_9, e_4, and e_5, respectively. ∎

The dual graph of a map of any graph on the plane is always a planar graph, though we will not prove this fact. What we will prove is that the chromatic number of a map, when it exists, is always equal to the vertex chromatic number of the map's dual graph. In this way many questions concerning coloring maps become questions about coloring graphs.

Proposition 12.75. *If $M_{\mathfrak{c}}(G)$ is a map of a graph G on the plane, then $\chi(M_{\mathfrak{c}}(G)) = \chi(G^*)$.*

Proof. The map $M_{\mathfrak{c}}(G)$ has a face that meets itself if and only if the dual graph G^* has a loop, so that $M_{\mathfrak{c}}(G)$ is uncolorable if and only if G^* is uncolorable, and thus $\chi(M_{\mathfrak{c}}(G))$ is undefined if and only if $\chi(G^*)$ is undefined.

Suppose that $M_{\mathfrak{c}}(G)$ has no face that meets itself, so that G^* is loopless and both $\chi(M_{\mathfrak{c}}(G))$ and $\chi(G^*)$ are defined. Let $\mathfrak{c}$ be a chromatic coloring of $M_{\mathfrak{c}}(G)$. Since two faces f_1 and f_2 in $M_{\mathfrak{c}}(G)$ meet if and only if vertices f_1^* and f_2^* in G^* are adjacent, if each $f^* \in V(G^*)$ is assigned the same color as $f \in F_{\mathfrak{c}}(G)$, then a proper vertex $\chi(M_{\mathfrak{c}}(G))$-coloring of G^* results, and we find that $\chi(G^*) \leq \chi(M_{\mathfrak{c}}(G))$. On the other hand, if $\mathfrak{c}$ is a chromatic coloring of G^*, then assigning each $f \in F_{\mathfrak{c}}(G)$ the same color as $f^* \in V(G^*)$ yields a proper map $\chi(G^*)$-coloring of $M_{\mathfrak{c}}(G)$, and so $\chi(M_{\mathfrak{c}}(G)) \leq \chi(G^*)$. Therefore $\chi(M_{\mathfrak{c}}(G)) = \chi(G^*)$. ■

Example 12.76. Let G be the graph with planar drawing $\mathfrak{c}(G)$ shown in Figure 34(a). The corresponding map $M_{\mathfrak{c}}(G)$, with faces labeled f_1 through f_8, is in Figure 34(b). To determine the chromatic number of the map, $\chi(M_{\mathfrak{c}}(G))$, we pass to the dual graph G^* in Figure 34(c) and endeavor to ascertain $\chi(G^*)$. The set of dual vertices is $V(G^*) = \{f_k^* : 1 \leq k \leq 8\}$, and, defining $E_4 = \{f_4^* f_k^* : k = 1, 2, 3, 5, 6, 7\}$ and $E_8 = \{f_8^* f_k^* : k = 1, 2, 3, 5, 7\}$, the set of dual edges is

$$E(G^*) = E_4 \cup E_8 \cup \{f_1^* f_2^*, f_1^* f_2^*, f_1^* f_2^*, f_1^* f_2^*, f_1^* f_2^*, f_1^* f_2^*\}.$$

The largest independent sets of vertices in G^* are $\{f_1^*, f_3^*, f_6^*\}$ and $\{f_2^*, f_5^*, f_7^*\}$. (We observe that the vertices in G^* with the two greatest degrees, f_4^* and f_8^*, are in neither of these sets.) Hence $\alpha(G^*) = 3$, and by Proposition 12.54 we have

$$\chi(G^*) \geq \left\lceil \frac{|V(G^*)|}{\alpha(G^*)} \right\rceil = \left\lceil \frac{8}{3} \right\rceil = 3.$$

To show that $\chi(G^*) = 3$, we need only find a proper vertex 3-coloring of G^*. Define the vertex 3-coloring $\mathfrak{c} : V(G) \to \{1, 2, 3\}$ as follows: $\mathfrak{c}(f_1^*) = 1$, $\mathfrak{c}(f_2^*) = 2$, $\mathfrak{c}(f_3^*) = 1$, $\mathfrak{c}(f_4^*) = 3$, $\mathfrak{c}(f_5^*) = 2$,

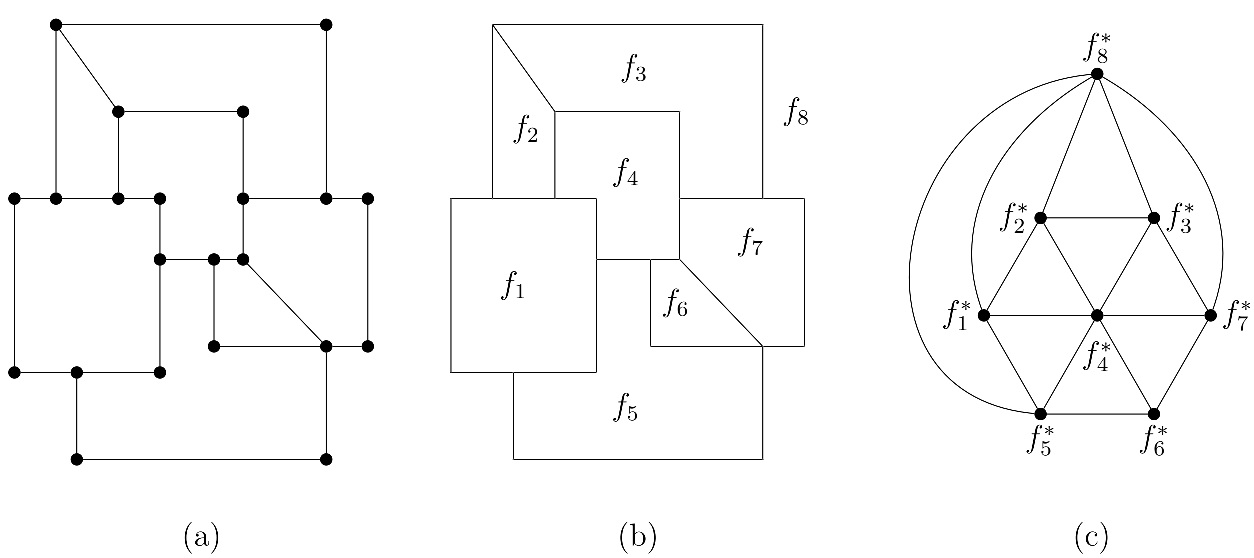
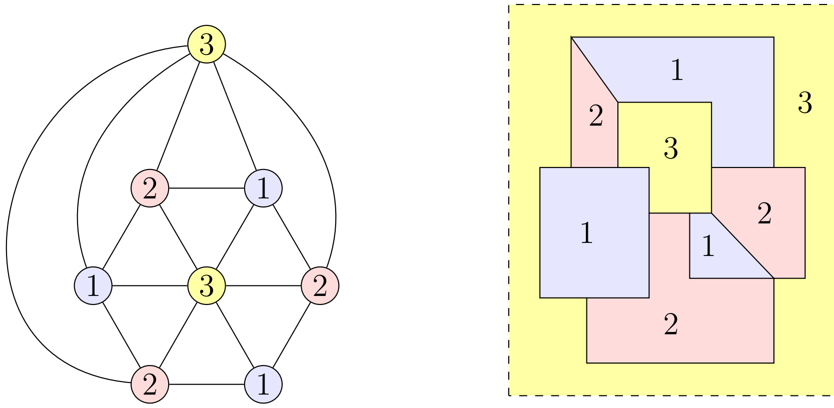


Figure 34.



(a) (b)

FIGURE 35.

$\mathfrak{c}(f_6^*) = 1$, $\mathfrak{c}(f_7^*) = 2$, $\mathfrak{c}(f_8^*) = 3$, shown in Figure 35(a). That this vertex 3-coloring is proper is evident by inspection, which establishes that $\chi(G^*) = 3$, and therefore $\chi(M_{\mathfrak{c}}(G)) = 3$ by Proposition 12.75. From the 3-coloring of G^* we readily obtain a proper map 3-coloring of $M_{\mathfrak{c}}(G)$, illustrated in Figure 35(b).

We make one last remark. In Figure 34(c) it can be seen that the subgraph $G^* - f_8^*$ is the even wheel W_6, and since $\chi(W_6) = 3$ by Proposition 12.61, it follows by Proposition 12.49 that $\chi(G^*) \geq 3$, as we had determined above by other means. ■

The objective of much of the remainder of this section will be to prove the five-color theorem, which states that the chromatic number of any simple planar graph is at most 5. Since the theorem does not require that a simple planar graph be connected, we need the following proposition in order to construct a technically airtight proof.

Proposition 12.77. *If $G_1, \ldots, G_k$ are the connected components of a simple graph G, then $\chi(G) = \max\{\chi(G_i) : 1 \leq i \leq k\}$.*

Proof. Suppose $G_1, \ldots, G_k$ are the connected components of a simple graph G, and let G_m be a component for which $\chi(G_m) = \max\{\chi(G_i) : 1 \leq i \leq k\}$. Since no edge in G has endpoints that lie in two different components, a chromatic coloring of each of the components results in a proper vertex coloring of G, and thus $\chi(G) \leq \chi(G_m)$. On the other hand, $\chi(G) \geq \chi(G_m)$ by Proposition 12.49. Therefore $\chi(G) = \chi(G_m)$. ■

Also required for the proof of the five-color theorem is a result from topology known as the Jordan curve theorem. The statement of the theorem may seem to be intuitively obvious, but the proof is surprisingly technical and so is omitted here.

Theorem 12.78 (**Jordan Curve Theorem**). *Any closed curve C in the plane separates the plane into two disjoint regions each having C as its boundary.*

Finally, the proof of the five-color theorem makes use of two graphical constructions known as an $\{i, j\}$-subgraph and a Kempe i-j chain. In the following definition recall that V_i denotes the color class of a color i (see Definition 12.46).

Definition 12.79. *Let i and j be two colors used in a vertex coloring of a graph G. The* $\{i,j\}$***-subgraph*** *of G is the subgraph of G induced by $V_i \cup V_j$. Any connected component of the $\{i,j\}$-subgraph is called a* ***Kempe*** i**-**j ***chain****.*

At last we state and prove the five-color theorem, which was first proved by Percy John Heawood way back in 1890 in the course of picking apart Alfred Kempe's 1879 "proof" of a four-color theorem. We shall remark on the four-color theorem at the end of this section.

Theorem 12.80 (Five-Color Theorem). *If G is a simple planar graph, then $\chi(G) \le 5$.*

Proof. If the theorem is true for any connected simple planar graph, then it is true for the connected components of any simple planar graph that is not connected, whereupon Proposition 12.77 implies it must hold true for the graph as a whole. Also, if the theorem is true for any chromatically critical simple planar graph, then it is true for any simple planar graph that is not chromatically critical, since edges and vertices may be removed from such a graph until a chromatically critical subgraph remains that has the same chromatic number as the original graph. Therefore, without loss of generality, we assume that G is a chromatically critical connected simple planar graph. To prove the theorem it is sufficient to demonstrate that G is vertex 5-colorable.

By Proposition 12.26 there exists some vertex $w \in V(G)$ such that $d_G(w) \le 5$, while Theorem 12.69 informs us that $d_G(v) \ge \chi(G) - 1$ for all $v \in V(G)$. Thus $\chi(G) \le d_G(w) + 1 \le 6$, and since G is chromatically critical, Proposition 12.68 implies that $\chi(G - w) = \chi(G) - 1 \le 5$; that is, $G - w$ is 5-colorable. Now, if $\mathfrak{c}$ is a proper 5-coloring of $G - w$, and not all five colors of $\mathfrak{c}$ have been assigned to the neighbors of w, then $\mathfrak{c}$ can be extended to a proper 5-coloring of G itself and there is nothing left to prove. We assume, therefore, that all five colors in the color set $\{1, 2, 3, 4, 5\}$ of $\mathfrak{c}$ are assigned to the vertices in $N_G(w)$, so that $d_G(w) = 5$. Furthermore, we contrive that the drawing function $\mathfrak{e}$ and coloring function $\mathfrak{c}$ be such that the planar drawing $\mathfrak{e}(G)$ puts the two neighbors of w colored 1 and 5 in one half-plane, the one colored 3 in the other half plane, and the two colored 2 and 4 on the line forming the boundary between the half-planes, as in Figure 36.

Let v_k denote the neighbor of w originally assigned the color k. The $\{2, 4\}$-subgraph of G will necessarily contain both v_2 and v_4. Let K be the Kempe 2-4 chain that contains v_2. There are two cases to consider: either K contains v_4 or it does not. We argue that in either case it will be possible to extend $\mathfrak{c}$ to a proper 5-coloring of G.

Case 1. Suppose K does not contain v_4, as illustrated at left in Figure 37. Then the colors 2 and 4 assigned to the vertices in K may be interchanged, or swapped, to obtain a new proper

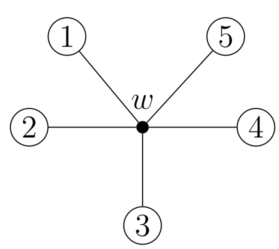


FIGURE 36.

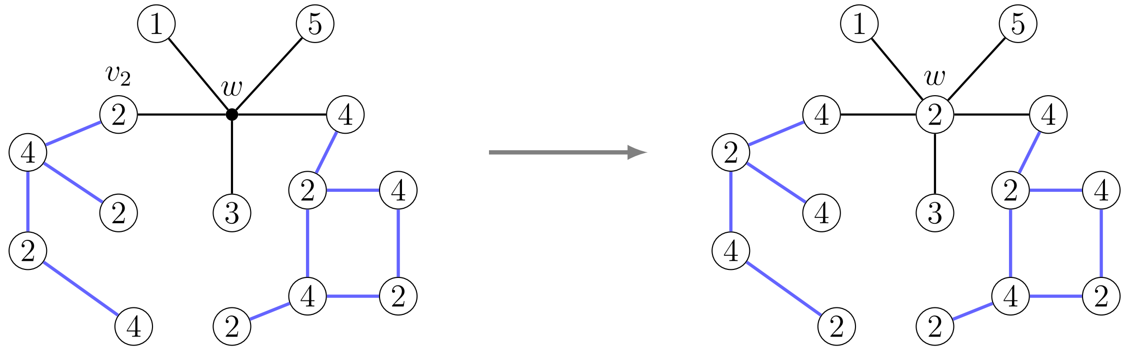


FIGURE 37. Swapping colors 2 and 4 in the Kempe 2-4 chain K.

5-coloring of $G - w$ in which no neighbor of w is assigned the color 2, and thus w may be assigned color 2 as at right in Figure 37 so as to obtain a proper 5-coloring of G.

Case 2. Suppose K contains both v_2 and v_4. Then by Proposition 10.72 there exists a v_2, v_4-path in K, as illustrated by the dotted edges at left in Figure 38. This path may be extended to become a cycle by concatenating it with the path having vertex sequence v_4wv_2, as the dotted edges at right in Figure 38 demonstrate. In the planar drawing $\mathfrak{e}(G)$ the cycle forms a closed curve that, by the Jordan curve theorem, separates the plane into two disjoint regions. One of these regions contains the (image of) vertex v_3, whilst the other contains the (images of) vertices v_1 and v_5. As a result, the Kempe 1-3 chain K' containing v_1 cannot also contain v_3 without the drawing $\mathfrak{e}(G)$ necessarily featuring an edge-crossing, which would contradict the hypothesis that $\mathfrak{e}(G)$ is a planar drawing. Thus the colors 1 and 3 assigned to the vertices of K' may be interchanged, so that v_1 is instead assigned the color 3, whereupon a proper 5-coloring of G may be obtained by assigned w the color 1. ■

As stated earlier, if a map $M_{\mathfrak{e}}(G)$ on the plane has no face that meets itself, then the dual graph G^* will be a simple planar graph. This in turn implies, by dint of the five-color theorem, that $\chi(G^*) \leq 5$, and hence $\chi(M_{\mathfrak{e}}(G)) \leq 5$ by Proposition 12.75. This leads to a map version of the five-color theorem.

Theorem 12.81 (**Five-Color Map Theorem**). *Any map of a graph on the plane that has no face that meets itself may be properly colored using no more than 5 colors.*

The five-color theorem is not the final word on the subject of coloring simple planar graphs, because in 1976 a proof for a four-color theorem was crafted by Kenneth Appel and Wolfgang Haken which has since held up to scrutiny and been improved upon. The proof was done with the aid of a supercomputer,[17] analyzing nearly 2000 cases, and so was ill-suited for a carbon-based readership. This generated no small amount of controversy, with some mathematicians not recognizing the work as a valid proof since it could not be humanly verified. In 1997 a team of researchers reduced the number of cases by about two-thirds, to 633. We present the theorem here, but the author must be forgiven for omitting the proof!

Theorem 12.82 (**Four-Color Theorem**). *If G is a simple planar graph, then $\chi(G) \leq 4$.*

[17]We remark that the supercomputers of the 1970s were several orders of magnitude slower than the average desktop computer of the 2020s.

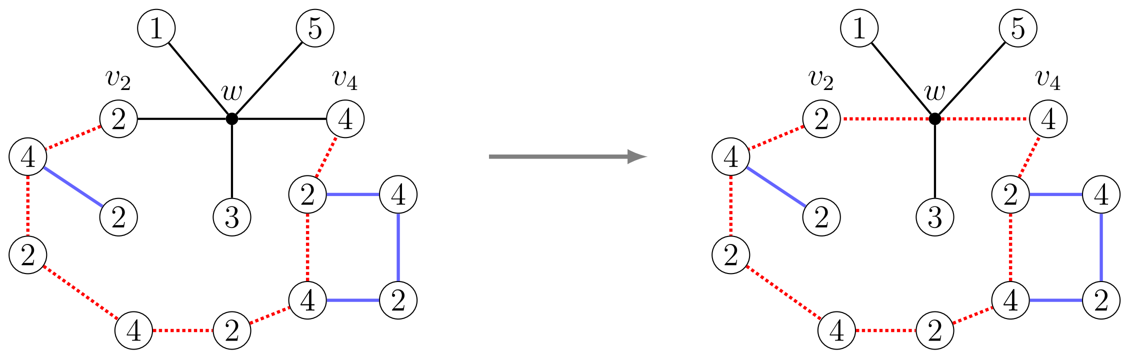


Figure 38. Forming a cycle in a Kempe 2-4 chain by extending a path.

As with the five-color theorem, there is an analog that applies to the coloring of maps rather than vertices. It follows from the four-color theorem and Proposition 12.75.

Theorem 12.83 (Four-Color Map Theorem). *Any map of a graph on the plane that has no face that meets itself may be properly colored using no more than 4 colors.*