

# Graphs

- 9.1 Graphs and Graph Models
- 9.2 Graph Terminology and Special Types of Graphs
- 9.3 Representing Graphs and Graph Isomorphism
- 9.4 Connectivity
- 9.5 Euler and Hamilton Paths
- 9.6 Shortest-Path Problems
- 9.7 Planar Graphs
- 9.8 Graph Coloring

**G**raphs are discrete structures consisting of vertices and edges that connect these vertices. Problems in almost every conceivable discipline can be solved using graph models. We will give examples to show how graphs are used as models in a variety of areas. For instance, we will show how graphs are used to represent the competition of different species in an ecological niche, how graphs are used to represent who influences whom in an organization, and how graphs are used to represent the outcomes of round-robin tournaments. We will describe how graphs can be used to model acquaintanceships between people, collaboration between researchers, telephone calls between telephone numbers, and links between websites. We will show how graphs can be used to model roadmaps and the assignment of jobs to employees of an organization.

Using graph models, we can determine whether it is possible to walk down all the streets in a city without going down a street twice, and we can find the number of colors needed to color the regions of a map. Graphs can be used to determine whether a circuit can be implemented on a planar circuit board. We can distinguish between two chemical compounds with the same molecular formula but different structures using graphs. We can determine whether two computers are connected by a communications link using graph models of computer networks. Graphs with weights assigned to their edges can be used to solve problems such as finding the shortest path between two cities in a transportation network. We can also use graphs to schedule exams and assign channels to television stations. This chapter will introduce the basic concepts of graph theory and present many different graph models. To solve the wide variety of problems that can be studied using graphs, we will introduce many different graph algorithms. We will also study the complexity of these algorithms.

## 9.1 Graphs and Graph Models

---

We begin with the definition of a graph.

**DEFINITION 1** A *graph*  $G = (V, E)$  consists of  $V$ , a nonempty set of *vertices* (or *nodes*) and  $E$ , a set of *edges*. Each edge has either one or two vertices associated with it, called its *endpoints*. An edge is said to *connect* its endpoints.

**Remark:** The set of vertices  $V$  of a graph  $G$  may be infinite. A graph with an infinite vertex set is called an **infinite graph**, and in comparison, a graph with a finite vertex set is called a **finite graph**. In this book we will usually consider only finite graphs.

Now suppose that a network is made up of data centers and communication links between computers. We can represent the location of each data center by a point and each communications link by a line segment, as shown in Figure 1.

This computer network can be modeled using a graph in which the vertices of the graph represent the data centers and the edges represent communication links. In general, we visualize graphs by using points to represent vertices and line segments, possibly curved, to represent edges, where the endpoints of a line segment representing an edge are the points representing the

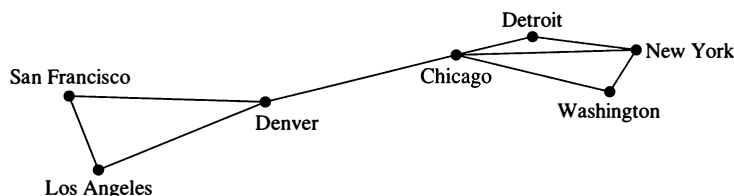


FIGURE 1 A Computer Network.

endpoints of the edge. When we draw a graph, we generally try to draw edges so that they do not cross. However, this is not necessary because any depiction using points to represent vertices and any form of connection between vertices can be used. Indeed, there are some graphs that cannot be drawn in the plane without edges crossing (see Section 9.7). The key point is that the way we draw a graph is arbitrary, as long as the correct connections between vertices are depicted.

Note that each edge of the graph representing this computer network connects two different vertices. That is, no edge connects a vertex to itself. Furthermore, no two different edges connect the same pair of vertices. A graph in which each edge connects two different vertices and where no two edges connect the same pair of vertices is called a **simple graph**. Note that in a simple graph, each edge is associated to an unordered pair of vertices, and no other edge is associated to this same edge. Consequently, when there is an edge of a simple graph associated to  $\{u, v\}$ , we can also say, without possible confusion, that  $\{u, v\}$  is an edge of the graph.

A computer network may contain multiple links between data centers, as shown in Figure 2. To model such networks we need graphs that have more than one edge connecting the same pair of vertices. Graphs that may have **multiple edges** connecting the same vertices are called **multigraphs**. When there are  $m$  different edges associated to the same unordered pair of vertices  $\{u, v\}$ , we also say that  $\{u, v\}$  is an edge of multiplicity  $m$ . That is, we can think of this set of edges as  $m$  different copies of an edge  $\{u, v\}$ .

Sometimes a communications link connects a data center with itself, perhaps a feedback loop for diagnostic purposes. Such a network is illustrated in Figure 3. To model this network we need to include edges that connect a vertex to itself. Such edges are called **loops**. Graphs that may include loops, and possibly multiple edges connecting the same pair of vertices, are sometimes called **pseudographs**.

So far the graphs we have introduced are **undirected graphs**. Their edges are also said to be **undirected**. However, to construct a graph model, we may find it necessary to assign directions to the edges of a graph. For example, in a computer network, some links may operate in only one direction (such links are called single duplex lines). This may be the case if there is a large amount of traffic sent to some data centers, with little or no traffic going in the opposite direction. Such a network is shown in Figure 4.

To model such a computer network we use a directed graph. Each edge of a directed graph is associated to an ordered pair. The definition of directed graph we give here is more general than the one we used in Chapter 8, where we used directed graphs to represent relations.

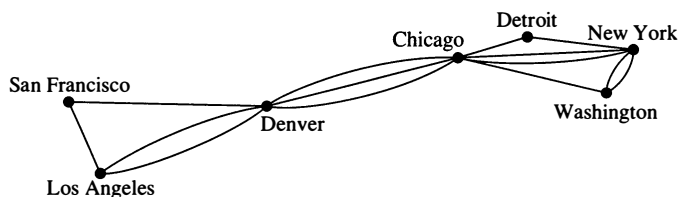


FIGURE 2 A Computer Network with Multiple Links between Data Centers.

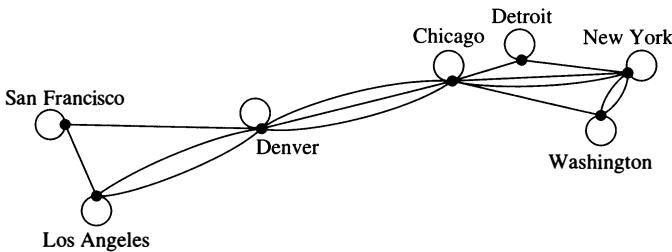


FIGURE 3 A Computer Network with Diagnostic Links.

**DEFINITION 2** A *directed graph* (or *digraph*)  $(V, E)$  consists of a nonempty set of vertices  $V$  and a set of *directed edges* (or *arcs*)  $E$ . Each directed edge is associated with an ordered pair of vertices. The directed edge associated with the ordered pair  $(u, v)$  is said to *start* at  $u$  and *end* at  $v$ .

When we depict a directed graph with a line drawing, we use an arrow pointing from  $u$  to  $v$  to indicate the direction of an edge that starts at  $u$  and ends at  $v$ . A directed graph may contain loops and it may contain multiple directed edges that start and end at the same vertices. A directed graph may also contain directed edges that connect vertices  $u$  and  $v$  in both directions; that is, when a digraph contains an edge from  $u$  to  $v$ , it may also contain one or more edges from  $v$  to  $u$ . Note that we obtain a directed graph when we assign a direction to each edge in an undirected graph. When a directed graph has no loops and has no multiple directed edges, it is called a **simple directed graph**. Because a simple directed graph has at most one edge associated to each ordered pair of vertices  $(u, v)$ , we call  $(u, v)$  an edge if there is an edge associated to it in the graph.

In some computer networks, multiple communication links between two data centers may be present, as illustrated in Figure 5. Directed graphs that may have **multiple directed edges** from a vertex to a second (possibly the same) vertex are used to model such networks. We call such graphs **directed multigraphs**. When there are  $m$  directed edges, each associated to an ordered pair of vertices  $(u, v)$ , we say that  $(u, v)$  is an edge of **multiplicity  $m$** .

For some models we may need a graph where some edges are undirected, while others are directed. A graph with both directed and undirected edges is called a **mixed graph**. For example, a mixed graph might be used to model a computer network containing links that operate in both directions and other links that operate only in one direction.

This terminology for the various types of graphs is summarized in Table 1. We will sometimes use the term **graph** as a general term to describe graphs with directed or undirected edges (or both), with or without loops, and with or without multiple edges. At other times, when the context is clear, we will use the term graph to refer only to undirected graphs.



Because of the relatively modern interest in graph theory, and because it has applications to a wide variety of disciplines, many different terminologies of graph theory have been introduced. The reader should determine how such terms are being used whenever they are encountered. The

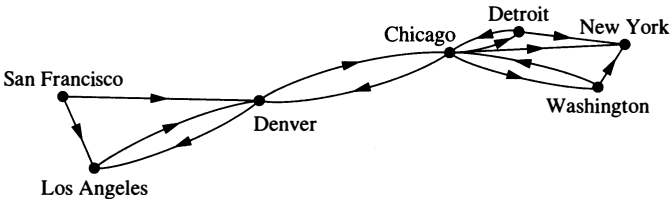


FIGURE 4 A Communications Network with One-Way Communications Links.

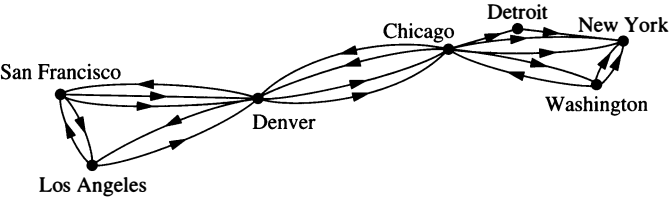


FIGURE 5 A Computer Network with Multiple One-Way Links.

terminology used by mathematicians to describe graphs has been increasingly standardized, but the terminology used to discuss graphs when they are used in other disciplines is still quite varied.

Although the terminology used to describe graphs may vary, three key questions can help us understand the structure of a graph:

- Are the edges of the graph undirected or directed (or both)?
- If the graph is undirected, are multiple edges present that connect the same pair of vertices? If the graph is directed, are multiple directed edges present?
- Are loops present?

Answering such questions helps us understand graphs. It is less important to remember the particular terminology used.

### Graph Models



Graphs are used in a wide variety of models. We will present a few graph models from diverse fields here. Others will be introduced in subsequent sections of this and later chapters. When we build a graph model, we need to make sure that we have correctly answered the three key questions we posed about the structure of a graph.

**EXAMPLE 1 Niche Overlap Graphs in Ecology** Graphs are used in many models involving the interaction of different species of animals. For instance, the competition between species in an ecosystem can be modeled using a **niche overlap graph**. Each species is represented by a vertex. An undirected edge connects two vertices if the two species represented by these vertices compete (that is, some of the food resources they use are the same). A niche overlap graph is a simple graph because no loops or multiple edges are needed in this model. The graph in Figure 6 models the ecosystem of a forest. We see from this graph that squirrels and raccoons compete but that crows and shrews do not. ◀



**EXAMPLE 2 Acquaintanceship Graphs** We can use graph models to represent various relationships between people. For example, we can use a simple graph to represent whether two people know



TABLE 1 Graph Terminology.			
Type	Edges	Multiple Edges Allowed?	Loops Allowed?
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

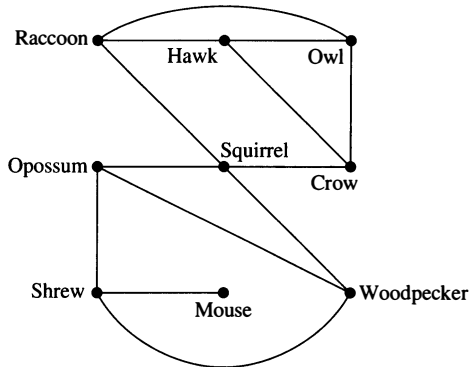


FIGURE 6 A Niche Overlap Graph.

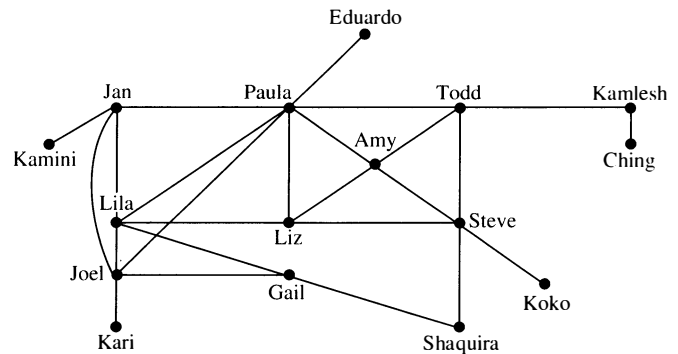


FIGURE 7 An Acquaintanceship Graph.

each other, that is, whether they are acquainted. Each person in a particular group of people is represented by a vertex. An undirected edge is used to connect two people when these people know each other. No multiple edges and usually no loops are used. (If we want to include the notion of self-knowledge, we would include loops.) A small acquaintanceship graph is shown in Figure 7. The acquaintanceship graph of all people in the world has more than six billion vertices and probably more than one trillion edges! We will discuss this graph further in Section 9.4. ◀

**EXAMPLE 3 Influence Graphs** In studies of group behavior it is observed that certain people can influence the thinking of others. A directed graph called an **influence graph** can be used to model this behavior. Each person of the group is represented by a vertex. There is a directed edge from vertex  $a$  to vertex  $b$  when the person represented by vertex  $a$  influences the person represented by vertex  $b$ . This graph does not contain loops and it does not contain multiple directed edges. An example of an influence graph for members of a group is shown in Figure 8. In the group modeled by this influence graph, Deborah can influence Brian, Fred, and Linda, but no one can influence her. Also, Yvonne and Brian can influence each other. ◀

**EXAMPLE 4 The Hollywood Graph** The **Hollywood graph** represents actors by vertices and connects two vertices when the actors represented by these vertices have worked together on a movie. This graph is a simple graph because its edges are undirected, it contains no multiple edges, and it contains no loops. According to the Internet Movie Database, in January 2006 the Hollywood graph had 637,099 vertices representing actors who have appeared in 339,896 films, and had more than 20 million edges. We will discuss some aspects of the Hollywood graph later in Section 9.4. ◀



**EXAMPLE 5 Round-Robin Tournaments** A tournament where each team plays each other team exactly once is called a **round-robin tournament**. Such tournaments can be modeled using directed graphs where each team is represented by a vertex. Note that  $(a, b)$  is an edge if team  $a$  beats team  $b$ . This graph is a simple directed graph, containing no loops or multiple directed edges (because no two teams play each other more than once). Such a directed graph model is presented in Figure 9. We see that Team 1 is undefeated in this tournament, and Team 3 is winless. ◀

**EXAMPLE 6 Collaboration Graphs** A graph called a **collaboration graph** can be used to model joint authorship of academic papers. In a collaboration graph, vertices represent people (perhaps restricted to members of a certain academic community) and edges link two people if they have jointly written a paper. This graph is a simple graph because it contains undirected edges and



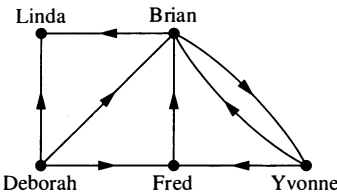


FIGURE 8 An Influence Graph.

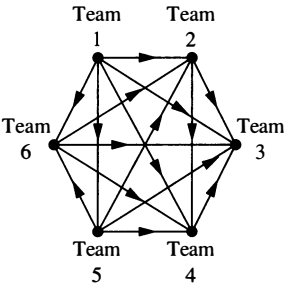


FIGURE 9 A Graph Model of a Round-Robin Tournament.

has no loops or multiple edges. The collaboration graph for people working together on research papers in mathematics has been found to have more than 400,000 vertices and 675,000 edges. We will have more to say about this graph in Section 9.4. ◀

**EXAMPLE 7**



**Call Graphs** Graphs can be used to model telephone calls made in a network, such as a long-distance telephone network. In particular, a directed multigraph can be used to model calls where each telephone number is represented by a vertex and each telephone call is represented by a directed edge. The edge representing a call starts at the telephone number from which the call was made and ends at the telephone number to which the call was made. We need directed edges because the direction in which the call is made matters. We need multiple directed edges because we want to represent each call made from a particular telephone number to a second number.

A small telephone call graph is displayed in Figure 10(a), representing seven telephone numbers. This graph shows, for instance, that three calls have been made from 732-555-1234 to 732-555-9876 and two in the other direction, but no calls have been made from 732-555-4444 to any of the other six numbers except 732-555-0011. When we care only whether there has been a call connecting two telephone numbers, we use an undirected graph with an edge connecting telephone numbers when there has been a call between these numbers. This version of the call graph is displayed in Figure 10(b).

Call graphs that model actual calling activities can be huge. For example, one call graph studied at AT&T, which models calls during 20 days, has about 290 million vertices and 4 billion edges. We will discuss call graphs further in Section 9.4. ◀

**EXAMPLE 8**



**The Web Graph** The World Wide Web can be modeled as a directed graph where each Web page is represented by a vertex and where an edge starts at the Web page  $a$  and ends at the Web page  $b$  if there is a link on  $a$  pointing to  $b$ . Because new Web pages are created and others

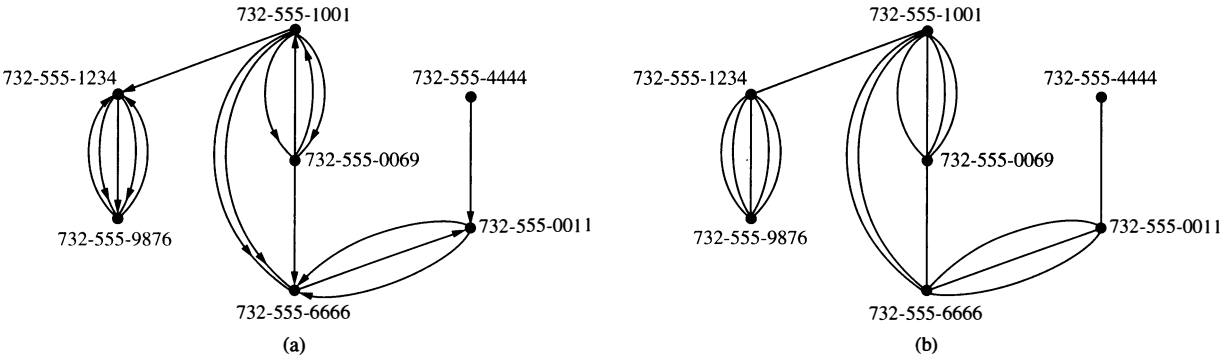


FIGURE 10 A Call Graph.

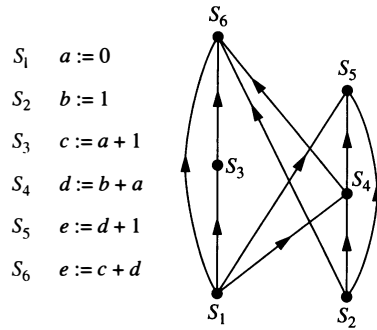


FIGURE 11 A Precedence Graph.

removed somewhere on the Web almost every second, the Web graph changes on an almost continual basis. Currently the Web graph has more than three billion vertices and 20 billion edges. Many people are studying the properties of the Web graph to better understand the nature of the Web. We will return to Web graphs in Section 9.4, and in Chapter 10 we will explain how the Web graph is used by the Web crawlers that search engines use to create indices of Web pages. ◀

**EXAMPLE 9 Precedence Graphs and Concurrent Processing** Computer programs can be executed more rapidly by executing certain statements concurrently. It is important not to execute a statement that requires results of statements not yet executed. The dependence of statements on previous statements can be represented by a directed graph. Each statement is represented by a vertex, and there is an edge from one vertex to a second vertex if the statement represented by the second vertex cannot be executed before the statement represented by the first vertex has been executed. This graph is called a **precedence graph**. A computer program and its graph are displayed in Figure 11. For instance, the graph shows that statement  $S_5$  cannot be executed before statements  $S_1$ ,  $S_2$ , and  $S_4$  are executed. ◀

**EXAMPLE 10 Roadmaps** Graphs can be used to model roadmaps. In such models, vertices represent intersections and edges represent roads. Undirected edges represent two-way roads and directed edges represent one-way roads. Multiple undirected edges represent multiple two-way roads connecting the same two intersections. Multiple directed edges represent multiple one-way roads that start at one intersection and end at a second intersection. Loops represent loop roads. Consequently, roadmaps depicting only two-way roads and no loop roads, and in which no two roads connect the same pair of intersections, can be represented using a simple undirected graph. Roadmaps depicting only one-way roads and no loop roads, and where no two roads start at the same intersection and end at the same intersection, can be modeled using simple directed graphs. Mixed graphs are needed to depict roadmaps that include both one-way and two-way roads. ◀

## Exercises

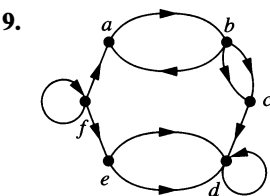
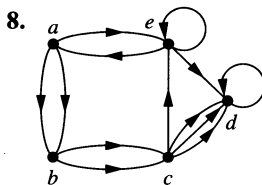
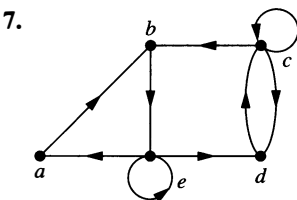
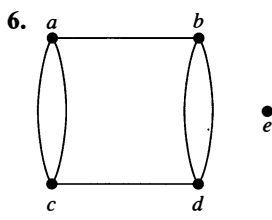
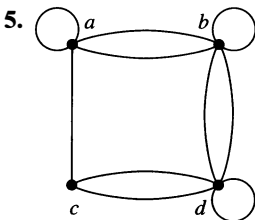
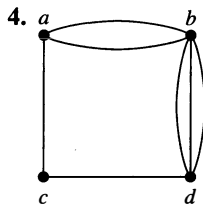
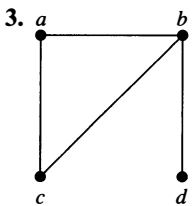
1. Draw graph models, stating the type of graph (from Table 1) used, to represent airline routes where every day there are four flights from Boston to Newark, two flights from Newark to Boston, three flights from Newark to Miami, two flights from Miami to Newark, one flight from Newark to Detroit, two flights from Detroit to Newark, three flights from Newark to Washington, two flights from

Washington to Newark, and one flight from Washington to Miami, with

- a) an edge between vertices representing cities that have a flight between them (in either direction).
- b) an edge between vertices representing cities for each flight that operates between them (in either direction).

- c) an edge between vertices representing cities for each flight that operates between them (in either direction), plus a loop for a special sightseeing trip that takes off and lands in Miami.
- d) an edge from a vertex representing a city where a flight starts to the vertex representing the city where it ends.
- e) an edge for each flight from a vertex representing a city where the flight begins to the vertex representing the city where the flight ends.
2. What kind of graph (from Table 1) can be used to model a highway system between major cities where
- a) there is an edge between the vertices representing cities if there is an interstate highway between them?
- b) there is an edge between the vertices representing cities for each interstate highway between them?
- c) there is an edge between the vertices representing cities for each interstate highway between them, and there is a loop at the vertex representing a city if there is an interstate highway that circles this city?

For Exercises 3–9, determine whether the graph shown has directed or undirected edges, whether it has multiple edges, and whether it has one or more loops. Use your answers to determine the type of graph in Table 1 this graph is.



10. For each undirected graph in Exercises 3–9 that is not simple, find a set of edges to remove to make it simple.
11. Let  $G$  be a simple graph. Show that the relation  $R$  on the set of vertices of  $G$  such that  $uRv$  if and only if there is an edge associated to  $\{u, v\}$  is a symmetric, irreflexive relation on  $G$ .
12. Let  $G$  be an undirected graph with a loop at every vertex. Show that the relation  $R$  on the set of vertices of  $G$  such that  $uRv$  if and only if there is an edge associated to  $\{u, v\}$  is a symmetric, reflexive relation on  $G$ .
13. The **intersection graph** of a collection of sets  $A_1, A_2, \dots, A_n$  is the graph that has a vertex for each of these sets and has an edge connecting the vertices representing two sets if these sets have a nonempty intersection. Construct the intersection graph of these collections of sets.
- a)  $A_1 = \{0, 2, 4, 6, 8\}$ ,  $A_2 = \{0, 1, 2, 3, 4\}$ ,  
 $A_3 = \{1, 3, 5, 7, 9\}$ ,  $A_4 = \{5, 6, 7, 8, 9\}$ ,  
 $A_5 = \{0, 1, 8, 9\}$
- b)  $A_1 = \{\dots, -4, -3, -2, -1, 0\}$ ,  
 $A_2 = \{\dots, -2, -1, 0, 1, 2, \dots\}$ ,  
 $A_3 = \{\dots, -6, -4, -2, 0, 2, 4, 6, \dots\}$ ,  
 $A_4 = \{\dots, -5, -3, -1, 1, 3, 5, \dots\}$ ,  
 $A_5 = \{\dots, -6, -3, 0, 3, 6, \dots\}$
- c)  $A_1 = \{x \mid x < 0\}$ ,  
 $A_2 = \{x \mid -1 < x < 0\}$ ,  
 $A_3 = \{x \mid 0 < x < 1\}$ ,  
 $A_4 = \{x \mid -1 < x < 1\}$ ,  
 $A_5 = \{x \mid x > -1\}$ ,  
 $A_6 = \mathbf{R}$
14. Use the niche overlap graph in Figure 6 to determine the species that compete with hawks.
15. Construct a niche overlap graph for six species of birds, where the hermit thrush competes with the robin and with the blue jay, the robin also competes with the mockingbird, the mockingbird also competes with the blue jay, and the nuthatch competes with the hairy woodpecker.
16. Draw the acquaintanceship graph that represents that Tom and Patricia, Tom and Hope, Tom and Sandy, Tom and Amy, Tom and Marika, Jeff and Patricia, Jeff and Mary, Patricia and Hope, Amy and Hope, and Amy and Marika know each other, but none of the other pairs of people listed know each other.
17. We can use a graph to represent whether two people were alive at the same time. Draw such a graph to represent whether each pair of the mathematicians and computer scientists with biographies in the first four chapters of this book who died before 1900 were contemporaneous. (Assume two people lived at the same time if they were alive during the same year.)
18. Who can influence Fred and whom can Fred influence in the influence graph in Example 3?
19. Construct an influence graph for the board members of a company if the President can influence the Director of Research and Development, the Director of Marketing,



and the Director of Operations; the Director of Research and Development can influence the Director of Operations; the Director of Marketing can influence the Director of Operations; and no one can influence, or be influenced by, the Chief Financial Officer.

20. Which other teams did Team 4 beat and which teams beat Team 4 in the round-robin tournament represented by the graph in Figure 9?
21. In a round-robin tournament the Tigers beat the Blue Jays, the Tigers beat the Cardinals, the Tigers beat the Orioles, the Blue Jays beat the Cardinals, the Blue Jays beat the Orioles, and the Cardinals beat the Orioles. Model this outcome with a directed graph.
22. Draw the call graph for the telephone numbers 555-0011, 555-1221, 555-1333, 555-8888, 555-2222, 555-0091, and 555-1200 if there were three calls from 555-0011 to 555-8888 and two calls from 555-8888 to 555-0011, two calls from 555-2222 to 555-0091, two calls from 555-1221 to each of the other numbers, and one call from 555-1333 to each of 555-0011, 555-1221, and 555-1200.
23. Explain how the two telephone call graphs for calls made during the month of January and calls made during the month of February can be used to determine the new telephone numbers of people who have changed their telephone numbers.
24. a) Explain how graphs can be used to model electronic mail messages in a network. Should the edges be directed or undirected? Should multiple edges be allowed? Should loops be allowed?  
b) Describe a graph that models the electronic mail sent in a network in a particular week.
25. How can a graph that models e-mail messages sent in a network be used to find people who have recently changed their primary e-mail address?
26. How can a graph that models e-mail messages sent in a network be used to find electronic mail mailing lists used to send the same message to many different e-mail addresses?
27. Describe a graph model that represents whether each person at a party knows the name of each other person at the party. Should the edges be directed or undirected? Should multiple edges be allowed? Should loops be allowed?
28. Describe a graph model that represents a subway system in a large city. Should edges be directed or undirected? Should multiple edges be allowed? Should loops be allowed?
29. Describe a graph model that represents traditional marriages between men and women. Does this graph have any special properties?
30. Which statements must be executed before  $S_6$  is executed in the program in Example 9? (Use the precedence graph in Figure 11.)
31. Construct a precedence graph for the following program:
 
$$\begin{aligned} S_1: x &:= 0 \\ S_2: x &:= x + 1 \\ S_3: y &:= 2 \\ S_4: z &:= y \\ S_5: x &:= x + 2 \\ S_6: y &:= x + z \\ S_7: z &:= 4 \end{aligned}$$
32. Describe a discrete structure based on a graph that can be used to model airline routes and their flight times. [Hint: Add structure to a directed graph.]
33. Describe a discrete structure based on a graph that can be used to model relationships between pairs of individuals in a group, where each individual may either like, dislike, or be neutral about another individual, and the reverse relationship may be different. [Hint: Add structure to a directed graph. Treat separately the edges in opposite directions between vertices representing two individuals.]
34. Describe a graph model that can be used to represent all forms of electronic communication between two people in a single graph. What kind of graph is needed?

## 9.2 Graph Terminology and Special Types of Graphs

### Introduction



We introduce some of the basic vocabulary of graph theory in this section. We will use this vocabulary later in this chapter when we solve many different types of problems. One such problem involves determining whether a graph can be drawn in the plane so that no two of its edges cross. Another example is deciding whether there is a one-to-one correspondence between the vertices of two graphs that produces a one-to-one correspondence between the edges of the graphs. We will also introduce several important families of graphs often used as examples and in models. Several important applications will be described where these special types of graphs arise.

## Basic Terminology

First, we give some terminology that describes the vertices and edges of undirected graphs.

**DEFINITION 1** Two vertices  $u$  and  $v$  in an undirected graph  $G$  are called *adjacent* (or *neighbors*) in  $G$  if  $u$  and  $v$  are endpoints of an edge of  $G$ . If  $e$  is associated with  $\{u, v\}$ , the edge  $e$  is called *incident with* the vertices  $u$  and  $v$ . The edge  $e$  is also said to *connect*  $u$  and  $v$ . The vertices  $u$  and  $v$  are called *endpoints* of an edge associated with  $\{u, v\}$ .

To keep track of how many edges are incident to a vertex, we make the following definition.

**DEFINITION 2** The *degree of a vertex in an undirected graph* is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex. The degree of the vertex  $v$  is denoted by  $\deg(v)$ .

**EXAMPLE 1** What are the degrees of the vertices in the graphs  $G$  and  $H$  displayed in Figure 1?

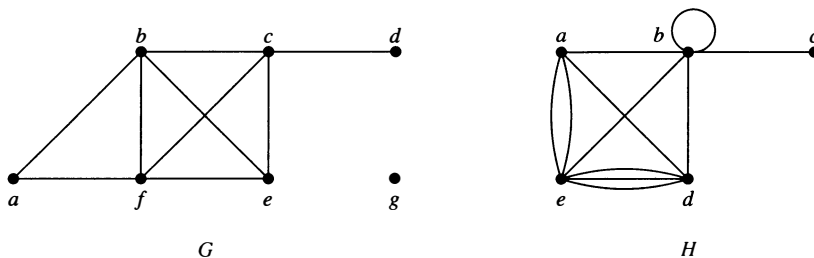
*Solution:* In  $G$ ,  $\deg(a) = 2$ ,  $\deg(b) = \deg(c) = \deg(f) = 4$ ,  $\deg(d) = 1$ ,  $\deg(e) = 3$ , and  $\deg(g) = 0$ . In  $H$ ,  $\deg(a) = 4$ ,  $\deg(b) = \deg(e) = 6$ ,  $\deg(c) = 1$ , and  $\deg(d) = 5$ . ◀

A vertex of degree zero is called **isolated**. It follows that an isolated vertex is not adjacent to any vertex. Vertex  $g$  in graph  $G$  in Example 1 is isolated. A vertex is **pendant** if and only if it has degree one. Consequently, a pendant vertex is adjacent to exactly one other vertex. Vertex  $d$  in graph  $G$  in Example 1 is pendant.

Examining the degrees of vertices in a graph model can provide useful information about the model, as Example 2 shows.

**EXAMPLE 2** What does the degree of a vertex in a niche overlap graph (introduced in Example 1 in Section 9.1) represent? Which vertices in this graph are pendant and which are isolated? Use the niche overlap graph shown in Figure 6 of Section 9.1 to interpret your answers.

*Solution:* There is an edge between two vertices in a niche overlap graph if and only if the two species represented by these vertices compete. Hence, the degree of a vertex in a niche overlap graph is the number of species in the ecosystem that compete with the species represented by this vertex. A vertex is pendant if the species competes with exactly one other species in the



**FIGURE 1** The Undirected Graphs  $G$  and  $H$ .

ecosystem. Finally, the vertex representing a species is isolated if this species does not compete with any other species in the ecosystem.

For instance, the degree of the vertex representing the squirrel in Figure 6 in Section 9.1 is four, because the squirrel competes with four other species: the crow, the opossum, the raccoon, and the woodpecker. In the graph in Figure 6, the mouse is the only species represented by a pendant vertex, because the mouse competes only with the shrew and all other species compete with at least two other species. The vertex representing a species is pendant if this species competes with only one other species. There are no isolated vertices in the graph in Figure 6 because every species in this ecosystem competes with at least one other species. ◀

What do we get when we add the degrees of all the vertices of a graph  $G = (V, E)$ ? Each edge contributes two to the sum of the degrees of the vertices because an edge is incident with exactly two (possibly equal) vertices. This means that the sum of the degrees of the vertices is twice the number of edges. We have the result in Theorem 1, which is sometimes called the Handshaking Theorem, because of the analogy between an edge having two endpoints and a handshake involving two hands.

**THEOREM 1 THE HANDSHAKING THEOREM** Let  $G = (V, E)$  be an undirected graph with  $e$  edges. Then

$$2e = \sum_{v \in V} \deg(v).$$

(Note that this applies even if multiple edges and loops are present.)

**EXAMPLE 3** How many edges are there in a graph with 10 vertices each of degree six?

*Solution:* Because the sum of the degrees of the vertices is  $6 \cdot 10 = 60$ , it follows that  $2e = 60$ . Therefore,  $e = 30$ . ◀

Theorem 1 shows that the sum of the degrees of the vertices of an undirected graph is even. This simple fact has many consequences, one of which is given as Theorem 2.

**THEOREM 2** An undirected graph has an even number of vertices of odd degree.

*Proof:* Let  $V_1$  and  $V_2$  be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph  $G = (V, E)$ . Then

$$2e = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v).$$

Because  $\deg(v)$  is even for  $v \in V_1$ , the first term in the right-hand side of the last equality is even. Furthermore, the sum of the two terms on the right-hand side of the last equality is even, because this sum is  $2e$ . Hence, the second term in the sum is also even. Because all the terms in this sum are odd, there must be an even number of such terms. Thus, there are an even number of vertices of odd degree. ◀

Terminology for graphs with directed edges reflects the fact that edges in directed graphs have directions.

**DEFINITION 3** When  $(u, v)$  is an edge of the graph  $G$  with directed edges,  $u$  is said to be *adjacent to*  $v$  and  $v$  is said to be *adjacent from*  $u$ . The vertex  $u$  is called the *initial vertex* of  $(u, v)$ , and  $v$  is called the *terminal* or *end vertex* of  $(u, v)$ . The initial vertex and terminal vertex of a loop are the same.

Because the edges in graphs with directed edges are ordered pairs, the definition of the degree of a vertex can be refined to reflect the number of edges with this vertex as the initial vertex and as the terminal vertex.

**DEFINITION 4** In a graph with directed edges the *in-degree* of a vertex  $v$ , denoted by  $\deg^-(v)$ , is the number of edges with  $v$  as their terminal vertex. The *out-degree* of  $v$ , denoted by  $\deg^+(v)$ , is the number of edges with  $v$  as their initial vertex. (Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of this vertex.)

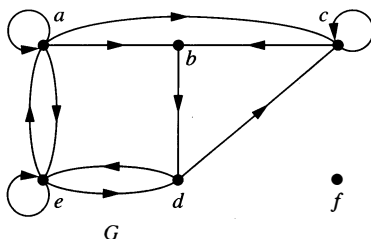
**EXAMPLE 4** Find the in-degree and out-degree of each vertex in the graph  $G$  with directed edges shown in Figure 2.

*Solution:* The in-degrees in  $G$  are  $\deg^-(a) = 2$ ,  $\deg^-(b) = 2$ ,  $\deg^-(c) = 3$ ,  $\deg^-(d) = 2$ ,  $\deg^-(e) = 3$ , and  $\deg^-(f) = 0$ . The out-degrees are  $\deg^+(a) = 4$ ,  $\deg^+(b) = 1$ ,  $\deg^+(c) = 2$ ,  $\deg^+(d) = 2$ ,  $\deg^+(e) = 3$ , and  $\deg^+(f) = 0$ . ◀

Because each edge has an initial vertex and a terminal vertex, the sum of the in-degrees and the sum of the out-degrees of all vertices in a graph with directed edges are the same. Both of these sums are the number of edges in the graph. This result is stated as Theorem 3.

**THEOREM 3** Let  $G = (V, E)$  be a graph with directed edges. Then

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



**FIGURE 2** The Directed Graph  $G$ .

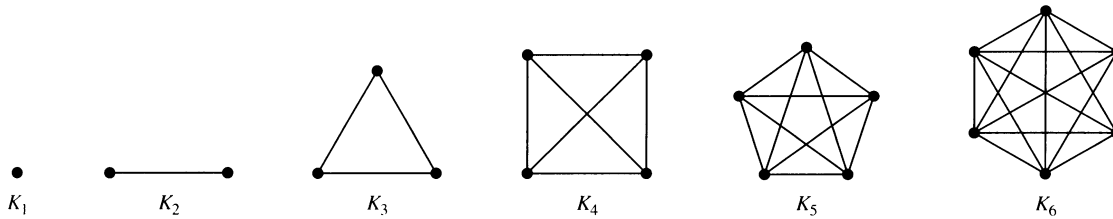


FIGURE 3 The Graphs  $K_n$  for  $1 \leq n \leq 6$ .

There are many properties of a graph with directed edges that do not depend on the direction of its edges. Consequently, it is often useful to ignore these directions. The undirected graph that results from ignoring directions of edges is called the **underlying undirected graph**. A graph with directed edges and its underlying undirected graph have the same number of edges.

### Some Special Simple Graphs

We will now introduce several classes of simple graphs. These graphs are often used as examples and arise in many applications.

**EXAMPLE 5 Complete Graphs** The **complete graph on  $n$  vertices**, denoted by  $K_n$ , is the simple graph that contains exactly one edge between each pair of distinct vertices. The graphs  $K_n$ , for  $n = 1, 2, 3, 4, 5, 6$ , are displayed in Figure 3. ◀

**EXAMPLE 6 Cycles** The **cycle  $C_n$** ,  $n \geq 3$ , consists of  $n$  vertices  $v_1, v_2, \dots, v_n$  and edges  $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}$ , and  $\{v_n, v_1\}$ . The cycles  $C_3, C_4, C_5$ , and  $C_6$  are displayed in Figure 4. ◀

**EXAMPLE 7 Wheels** We obtain the **wheel  $W_n$**  when we add an additional vertex to the cycle  $C_n$ , for  $n \geq 3$ , and connect this new vertex to each of the  $n$  vertices in  $C_n$ , by new edges. The wheels  $W_3, W_4, W_5$ , and  $W_6$  are displayed in Figure 5. ◀

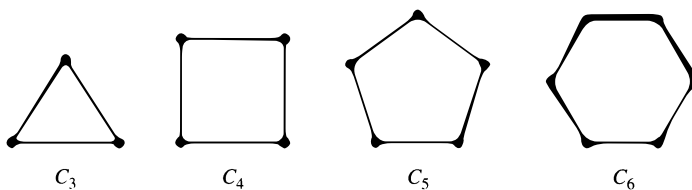


FIGURE 4 The Cycles  $C_3, C_4, C_5$ , and  $C_6$ .

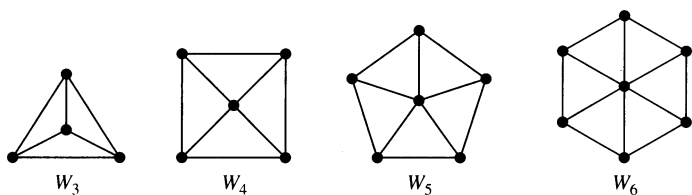


FIGURE 5 The Wheels  $W_3, W_4, W_5$ , and  $W_6$ .

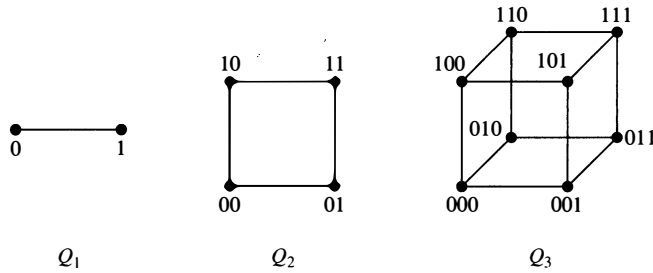


FIGURE 6 The  $n$ -cube  $Q_n$  for  $n = 1, 2$ , and  $3$ .

**EXAMPLE 8  $n$ -Cubes** The  $n$ -dimensional hypercube, or  $n$ -cube, denoted by  $Q_n$ , is the graph that has vertices representing the  $2^n$  bit strings of length  $n$ . Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position. The graphs  $Q_1$ ,  $Q_2$ , and  $Q_3$  are displayed in Figure 6. Note that you can construct the  $(n + 1)$ -cube  $Q_{n+1}$  from the  $n$ -cube  $Q_n$  by making two copies of  $Q_n$ , prefacing the labels on the vertices with a 0 in one copy of  $Q_n$  and with a 1 in the other copy of  $Q_n$ , and adding edges connecting two vertices that have labels differing only in the first bit. In Figure 6,  $Q_3$  is constructed from  $Q_2$  by drawing two copies of  $Q_2$  as the top and bottom faces of  $Q_3$ , adding 0 at the beginning of the label of each vertex in the bottom face and 1 at the beginning of the label of each vertex in the top face. (Here, by *face* we mean a face of a cube in three-dimensional space. Think of drawing the graph  $Q_3$  in three-dimensional space with copies of  $Q_2$  as the top and bottom faces of a cube and then drawing the projection of the resulting depiction in the plane.) ◀

## Bipartite Graphs



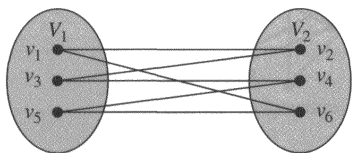
Sometimes a graph has the property that its vertex set can be divided into two disjoint subsets such that each edge connects a vertex in one of these subsets to a vertex in the other subset. For example, consider the graph representing marriages between men and women in a village, where each person is represented by a vertex and a marriage is represented by an edge. In this graph, each edge connects a vertex in the subset of vertices representing males and a vertex in the subset of vertices representing females. This leads us to Definition 5.

**DEFINITION 5** A simple graph  $G$  is called *bipartite* if its vertex set  $V$  can be partitioned into two disjoint sets  $V_1$  and  $V_2$  such that every edge in the graph connects a vertex in  $V_1$  and a vertex in  $V_2$  (so that no edge in  $G$  connects either two vertices in  $V_1$  or two vertices in  $V_2$ ). When this condition holds, we call the pair  $(V_1, V_2)$  a *bipartition* of the vertex set  $V$  of  $G$ .

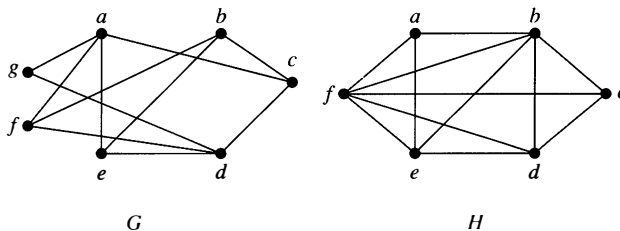
In Example 9 we will show that  $C_6$  is bipartite, and in Example 10 we will show that  $K_3$  is not bipartite.

**EXAMPLE 9**  $C_6$  is bipartite, as shown in Figure 7, because its vertex set can be partitioned into the two sets  $V_1 = \{v_1, v_3, v_5\}$  and  $V_2 = \{v_2, v_4, v_6\}$ , and every edge of  $C_6$  connects a vertex in  $V_1$  and a vertex in  $V_2$ . ◀

**EXAMPLE 10**  $K_3$  is not bipartite. To verify this, note that if we divide the vertex set of  $K_3$  into two disjoint sets, one of the two sets must contain two vertices. If the graph were bipartite, these two vertices



**FIGURE 7** Showing That  $C_6$  Is Bipartite.



**FIGURE 8** The Undirected Graphs  $G$  and  $H$ .

could not be connected by an edge, but in  $K_3$  each vertex is connected to every other vertex by an edge. ◀

**EXAMPLE 11** Are the graphs  $G$  and  $H$  displayed in Figure 8 bipartite?

*Solution:* Graph  $G$  is bipartite because its vertex set is the union of two disjoint sets,  $\{a, b, d\}$  and  $\{c, e, f, g\}$ , and each edge connects a vertex in one of these subsets to a vertex in the other subset. (Note that for  $G$  to be bipartite it is not necessary that every vertex in  $\{a, b, d\}$  be adjacent to every vertex in  $\{c, e, f, g\}$ . For instance,  $b$  and  $g$  are not adjacent.)

Graph  $H$  is not bipartite because its vertex set cannot be partitioned into two subsets so that edges do not connect two vertices from the same subset. (The reader should verify this by considering the vertices  $a$ ,  $b$ , and  $f$ .) ◀

Theorem 4 provides a useful criterion for determining whether a graph is bipartite.

**THEOREM 4** A simple graph is bipartite if and only if it is possible to assign one of two different colors to each vertex of the graph so that no two adjacent vertices are assigned the same color.

*Proof:* First, suppose that  $G = (V, E)$  is a bipartite simple graph. Then  $V = V_1 \cup V_2$ , where  $V_1$  and  $V_2$  are disjoint sets and every edge in  $E$  connects a vertex in  $V_1$  and a vertex in  $V_2$ . If we assign one color to each vertex in  $V_1$  and a second color to each vertex in  $V_2$ , then no two adjacent vertices are assigned the same color.

Now suppose that it is possible to assign colors to the vertices of the graph using just two colors so that no two adjacent vertices are assigned the same color. Let  $V_1$  be the set of vertices assigned one color and  $V_2$  be the set of vertices assigned the other color. Then,  $V_1$  and  $V_2$  are disjoint and  $V = V_1 \cup V_2$ . Furthermore, every edge connects a vertex in  $V_1$  and a vertex in  $V_2$  because no two adjacent vertices are either both in  $V_1$  or both in  $V_2$ . Consequently,  $G$  is bipartite. ◀

We illustrate how Theorem 4 can be used to determine whether a graph is bipartite in Example 12.

**EXAMPLE 12** Use Theorem 4 to determine whether the graphs in Example 11 are bipartite.

*Solution:* We first consider the graph  $G$ . We will try to assign one of two colors, say red and blue, to each vertex in  $G$  so that no edge in  $G$  connects a red vertex and a blue vertex. Without loss of generality we begin by arbitrarily assigning red to  $a$ . Then, we must assign blue to  $c$ ,  $e$ ,  $f$ , and  $g$ , because each of these vertices is adjacent to  $a$ . To avoid having an edge with two blue endpoints, we must assign red to all the vertices adjacent to either  $c$ ,  $e$ ,  $f$ , or  $g$ . This means that

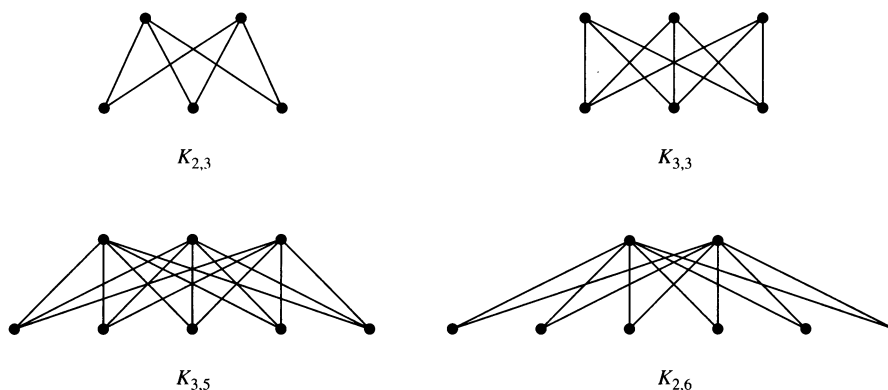


FIGURE 9 Some Complete Bipartite Graphs.

we must assign red to both  $b$  and  $d$  (and means that  $a$  must be assigned red, which it already has been). We have now assigned colors to all vertices, with  $a$ ,  $b$ , and  $d$  red and  $c$ ,  $e$ ,  $f$ , and  $g$  blue. Checking all edges, we see that every edge connects a red vertex and a blue vertex. Hence, by Theorem 4 the graph  $G$  is bipartite.

Next, we will try to assign either red or blue to each vertex in  $H$  so that no edge in  $H$  connects a red vertex and a blue vertex. Without loss of generality we arbitrarily assign red to  $a$ . Then, we must assign blue to  $b$ ,  $e$ , and  $f$ , because each is adjacent to  $a$ . But this is not possible because  $e$  and  $f$  are adjacent, so both cannot be assigned blue. This argument shows that we cannot assign one of two colors to each of the vertices of  $H$  so that no adjacent vertices are assigned the same color. It follows by Theorem 4 that  $H$  is not bipartite. ◀

Theorem 4 is an example of a result in the part of graph theory known as graph colorings. Graph colorings is an important part of graph theory with important applications. We will study graph colorings further in Section 9.8.

Another useful criterion for determining whether a graph is bipartite is based on the notion of a path, a topic we study in Section 9.4. A graph is bipartite if and only if it is not possible to start at a vertex and return to this vertex by traversing an odd number of distinct edges. We will make this notion more precise when we discuss paths and circuits in graphs in Section 9.4 (see Exercise 53 in that section).

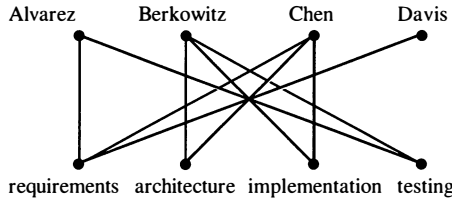
**EXAMPLE 13 Complete Bipartite Graphs** The complete bipartite graph  $K_{m,n}$  is the graph that has its vertex set partitioned into two subsets of  $m$  and  $n$  vertices, respectively. There is an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset. The complete bipartite graphs  $K_{2,3}$ ,  $K_{3,3}$ ,  $K_{3,5}$ , and  $K_{2,6}$  are displayed in Figure 9. ◀

## Some Applications of Special Types of Graphs

We will show how bipartite graphs and special types of graphs are used in models in Examples 14–16.

**EXAMPLE 14 Job Assignments** Suppose that there are  $m$  employees in a group and  $j$  different jobs that need to be done where  $m \leq j$ . Each employee is trained to do one or more of these  $j$  jobs. We can use a graph to model employee capabilities. We represent each employee by a vertex and each job by a vertex. For each employee, we include an edge from the vertex representing that employee to the vertices representing all jobs that the employee has been trained to do.





**FIGURE 10 Modeling the Jobs for Which Employees Have Been Trained.**

Note that the vertex set of this graph can be partitioned into two disjoint sets, the set of vertices representing employees and the set of vertices representing jobs, and each edge connects a vertex representing an employee to a vertex representing a job. Consequently, this graph is bipartite.

For example, suppose that a group has four employees: Alvarez, Berkowitz, Chen, and Davis; and suppose that four jobs need to be done to complete a project: requirements, architecture, implementation, and testing. Suppose that Alvarez has been trained to do requirements and testing; Berkowitz has been trained to do architecture, implementation, and testing; Chen has been trained to do requirements, architecture, and implementation; and Davis has only been trained to do requirements. We can model these capabilities of employees using the bipartite graph shown in Figure 10.

To complete the project, we must assign jobs to the employees so that every job has an employee assigned to it and no employee is assigned more than one job. In this case, we can assign Alvarez to do testing, Berkowitz to do implementation, Chen to do architecture, and Davis to do requirements, as shown in Figure 10 (where colored lines show this assignment of jobs). ◀

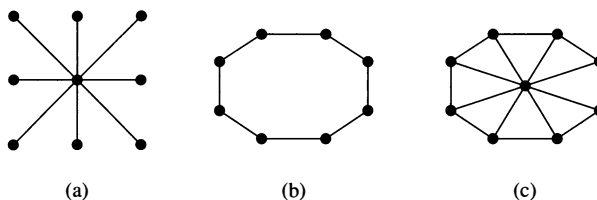
Finding an assignment of jobs to employees can be thought of as finding a matching in the graph model. A **matching** in a simple graph is a subset of the set of edges of the graph such that no two edges are incident with the same vertex; a **maximal matching** is a matching with the largest number of edges. In other words, a matching is a subset of edges such that if  $\{s, t\}$  and  $\{u, v\}$  are edges of the matching, then  $s, t, u$ , and  $v$  are distinct. To assign jobs to employees so that the largest number of employees are assigned jobs, we seek a maximum matching in the graph that models employee capabilities. (The interested reader can find more about matchings in books about graph theory, including [GrYe06].)

### EXAMPLE 15



**Local Area Networks** The various computers in a building, such as minicomputers and personal computers, as well as peripheral devices such as printers and plotters, can be connected using a *local area network*. Some of these networks are based on a *star topology*, where all devices are connected to a central control device. A local area network can be represented using a complete bipartite graph  $K_{1,n}$ , as shown in Figure 11(a). Messages are sent from device to device through the central control device.

Other local area networks are based on a *ring topology*, where each device is connected to exactly two others. Local area networks with a ring topology are modeled using  $n$ -cycles,  $C_n$ , as shown in Figure 11(b). Messages are sent from device to device around the cycle until the intended recipient of a message is reached.



**FIGURE 11 Star, Ring, and Hybrid Topologies for Local Area Networks.**

Finally, some local area networks use a hybrid of these two topologies. Messages may be sent around the ring, or through a central device. This redundancy makes the network more reliable. Local area networks with this redundancy can be modeled using wheels  $W_n$ , as shown in Figure 11(c). ◀

**EXAMPLE 16 Interconnection Networks for Parallel Computation** For many years, computers executed programs one operation at a time. Consequently, the algorithms written to solve problems were designed to perform one step at a time; such algorithms are called **serial**. (Almost all algorithms described in this book are serial.) However, many computationally intense problems, such as weather simulations, medical imaging, and cryptanalysis, cannot be solved in a reasonable amount of time using serial operations, even on a supercomputer. Furthermore, there is a physical limit to how fast a computer can carry out basic operations, so there will always be problems that cannot be solved in a reasonable length of time using serial operations.

**Parallel processing**, which uses computers made up of many separate processors, each with its own memory, helps overcome the limitations of computers with a single processor. **Parallel algorithms**, which break a problem into a number of subproblems that can be solved concurrently, can then be devised to rapidly solve problems using a computer with multiple processors. In a parallel algorithm, a single instruction stream controls the execution of the algorithm, sending subproblems to different processors, and directs the input and output of these subproblems to the appropriate processors.

When parallel processing is used, one processor may need output generated by another processor. Consequently, these processors need to be interconnected. We can use the appropriate type of graph to represent the interconnection network of the processors in a computer with multiple processors. In the following discussion, we will describe the most commonly used types of interconnection networks for parallel processors. The type of interconnection network used to implement a particular parallel algorithm depends on the requirements for exchange of data between processors, the desired speed, and, of course, the available hardware.

The simplest, but most expensive, network-interconnecting processors include a two-way link between each pair of processors. This network can be represented by  $K_n$ , the complete graph on  $n$  vertices, when there are  $n$  processors. However, there are serious problems with this type of interconnection network because the required number of connections is so large. In reality, the number of direct connections to a processor is limited, so when there are a large number of processors, a processor cannot be linked directly to all others. For example, when there are 64 processors,  $C(64, 2) = 2016$  connections would be required, and each processor would have to be directly connected to 63 others.

On the other hand, perhaps the simplest way to interconnect  $n$  processors is to use an arrangement known as a **linear array**. Each processor  $P_i$ , other than  $P_1$  and  $P_n$ , is connected to its neighbors  $P_{i-1}$  and  $P_{i+1}$  via a two-way link.  $P_1$  is connected only to  $P_2$ , and  $P_n$  is connected only to  $P_{n-1}$ . The linear array for six processors is shown in Figure 12. The advantage of a linear array is that each processor has at most two direct connections to other processors. The disadvantage is that it is sometimes necessary to use a large number of intermediate links, called **hops**, for processors to share information.

The **mesh network** (or **two-dimensional array**) is a commonly used interconnection network. In such a network, the number of processors is a perfect square, say  $n = m^2$ . The  $n$  processors are labeled  $P(i, j)$ ,  $0 \leq i \leq m - 1$ ,  $0 \leq j \leq m - 1$ . Two-way links connect processor  $P(i, j)$  with its four neighbors, processors  $P(i \pm 1, j)$  and  $P(i, j \pm 1)$ , as long as these are processors in the mesh. (Note that four processors, on the corners of the mesh, have only two adjacent processors, and other processors on the boundaries have only three neighbors. Sometimes a variant of a mesh network in which every processor has exactly four connections is used; see Exercise 66 at the end of this section.) The mesh network limits the number of links for each processor. Communication between some pairs of processors requires  $O(\sqrt{n}) = O(m)$

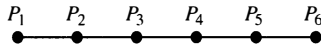


FIGURE 12 A Linear Array for Six Processors.

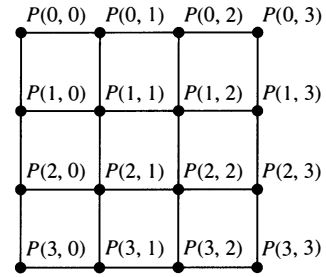


FIGURE 13 A Mesh Network for 16 Processors.

intermediate links. (See Exercise 67 at the end of this section.) The graph representing the mesh network for 16 processors is shown in Figure 13.

One important type of interconnection network is the hypercube. For such a network, the number of processors is a power of 2,  $n = 2^m$ . The  $n$  processors are labeled  $P_0, P_1, \dots, P_{n-1}$ . Each processor has two-way connections to  $m$  other processors. Processor  $P_i$  is linked to the processors with indices whose binary representations differ from the binary representation of  $i$  in exactly one bit. The hypercube network balances the number of direct connections for each processor and the number of intermediate connections required so that processors can communicate. Many computers have been built using a hypercube network, and many parallel algorithms have been devised that use a hypercube network. The graph  $Q_m$ , the  $m$ -cube, represents the hypercube network with  $n = 2^m$  processors. Figure 14 displays the hypercube network for eight processors. (Figure 14 displays a different way to draw  $Q_3$  than was shown in Figure 6.)

## New Graphs from Old

Sometimes we need only part of a graph to solve a problem. For instance, we may care only about the part of a large computer network that involves the computer centers in New York, Denver, Detroit, and Atlanta. Then we can ignore the other computer centers and all telephone lines not linking two of these specific four computer centers. In the graph model for the large network, we can remove the vertices corresponding to the computer centers other than the four of interest, and we can remove all edges incident with a vertex that was removed. When edges and vertices are removed from a graph, without removing endpoints of any remaining edges, a smaller graph is obtained. Such a graph is called a **subgraph** of the original graph.

**DEFINITION 6** A *subgraph* of a graph  $G = (V, E)$  is a graph  $H = (W, F)$ , where  $W \subseteq V$  and  $F \subseteq E$ . A subgraph  $H$  of  $G$  is a *proper subgraph* of  $G$  if  $H \neq G$ .

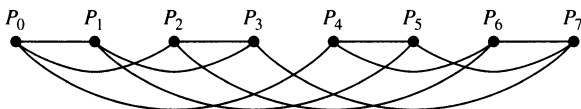


FIGURE 14 A Hypercube Network for Eight Processors.

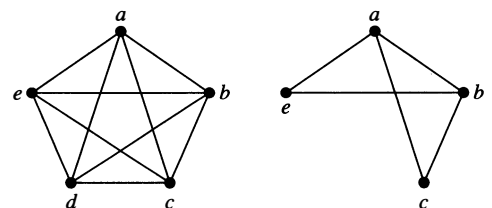


FIGURE 15 A Subgraph of  $K_5$ .

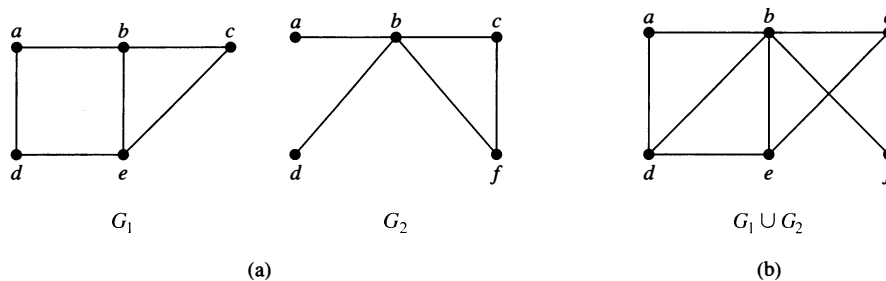


FIGURE 16 (a) The Simple Graphs  $G_1$  and  $G_2$ ; (b) Their Union  $G_1 \cup G_2$ .

**EXAMPLE 17** The graph  $G$  shown in Figure 15 is a subgraph of  $K_5$ .

Two or more graphs can be combined in various ways. The new graph that contains all the vertices and edges of these graphs is called the **union** of the graphs. We will give a more formal definition for the union of two simple graphs.

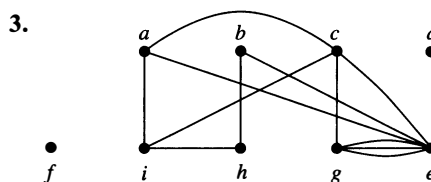
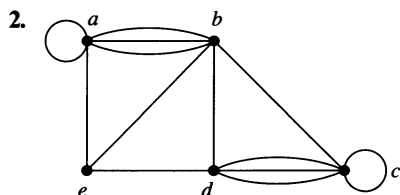
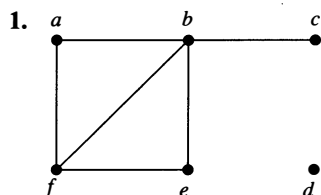
**DEFINITION 7** The *union* of two simple graphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  is the simple graph with vertex set  $V_1 \cup V_2$  and edge set  $E_1 \cup E_2$ . The union of  $G_1$  and  $G_2$  is denoted by  $G_1 \cup G_2$ .

**EXAMPLE 18** Find the union of the graphs  $G_1$  and  $G_2$  shown in Figure 16(a).

*Solution:* The vertex set of the union  $G_1 \cup G_2$  is the union of the two vertex sets, namely,  $\{a, b, c, d, e, f\}$ . The edge set of the union is the union of the two edge sets. The union is displayed in Figure 16(b).

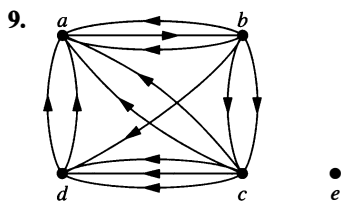
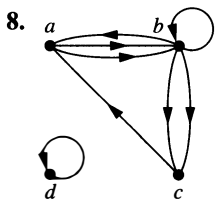
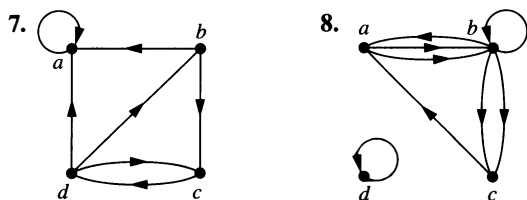
## Exercises

In Exercises 1–3 find the number of vertices, the number of edges, and the degree of each vertex in the given undirected graph. Identify all isolated and pendant vertices.



- Find the sum of the degrees of the vertices of each graph in Exercises 1–3 and verify that it equals twice the number of edges in the graph.
- Can a simple graph exist with 15 vertices each of degree five?
- Show that the sum, over the set of people at a party, of the number of people a person has shaken hands with, is even. Assume that no one shakes his or her own hand.

In Exercises 7–9 determine the number of vertices and edges and find the in-degree and out-degree of each vertex for the given directed multigraph.



10. For each of the graphs in Exercises 7–9 determine the sum of the in-degrees of the vertices and the sum of the out-degrees of the vertices directly. Show that they are both equal to the number of edges in the graph.

11. Construct the underlying undirected graph for the graph with directed edges in Figure 2.

12. What does the degree of a vertex represent in the acquaintanceship graph, where vertices represent all the people in the world? What do isolated and pendant vertices in this graph represent? In one study it was estimated that the average degree of a vertex in this graph is 1000. What does this mean in terms of the model?

13. What does the degree of a vertex represent in a collaboration graph? What do isolated and pendant vertices represent?

14. What does the degree of a vertex in the Hollywood graph represent? What do the isolated and pendant vertices represent?

15. What do the in-degree and the out-degree of a vertex in a telephone call graph, as described in Example 7 of Section 9.1, represent? What does the degree of a vertex in the undirected version of this graph represent?

16. What do the in-degree and the out-degree of a vertex in the Web graph, as described in Example 8 of Section 9.1, represent?

17. What do the in-degree and the out-degree of a vertex in a directed graph modeling a round-robin tournament represent?

18. Show that in a simple graph with at least two vertices there must be two vertices that have the same degree.

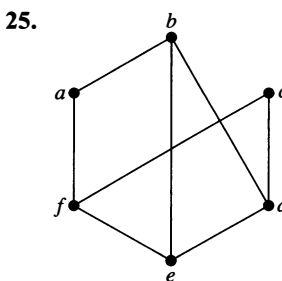
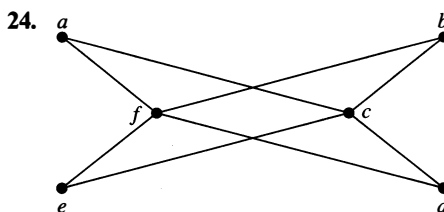
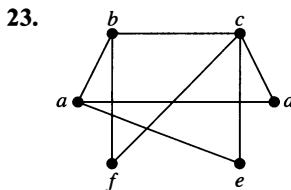
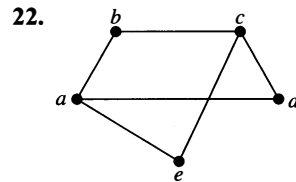
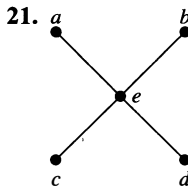
19. Use Exercise 18 to show that in a group, there must be two people who know the same number of other people in the group.

20. Draw these graphs.

- a)  $K_7$                       b)  $K_{1,8}$                       c)  $K_{4,4}$   
d)  $C_7$                       e)  $W_7$                       f)  $Q_4$

In Exercises 21–25 determine whether the graph is bipartite. You may find it useful to apply Theorem 4 and answer the question by determining whether it is possible to assign either

red or blue to each vertex so that no two adjacent vertices are assigned the same color.



26. For which values of  $n$  are these graphs bipartite?

- a)  $K_n$                       b)  $C_n$                       c)  $W_n$                       d)  $Q_n$

27. Suppose that a new company has five employees: Zamora, Agraharam, Smith, Chou, and Macintyre. Each employee will assume one of six responsibilities: planning, publicity, sales, marketing, development, and industry relations. Each employee is capable of doing one or more of these jobs: Zamora could do planning, sales, marketing, or industry relations; Agraharam could do planning or development; Smith could do publicity, sales, or industry relations; Chou could do planning, sales, or industry relations; and Macintyre could do planning, publicity, sales, or industry relations.

- a) Model the capabilities of these employees using a bipartite graph.  
b) Find an assignment of responsibilities such that each employee is assigned a responsibility.

28. Suppose that there are five young women and six young men on an island. Each woman is willing to marry some

of the men on the island and each man is willing to marry any woman who is willing to marry him. Suppose that Anna is willing to marry Jason, Larry, and Matt; Barbara is willing to marry Kevin and Larry; Carol is willing to marry Jason, Nick, and Oscar; Diane is willing to marry Jason, Larry, Nick, and Oscar; and Elizabeth is willing to marry Jason and Matt.

- a) Model the possible marriages on the island using a bipartite graph.
- b) Find a matching of the young women and the young men on the island such that each young woman is matched with a young man whom she is willing to marry.

29. How many vertices and how many edges do these graphs have?

- a)  $K_n$                       b)  $C_n$                       c)  $W_n$
- d)  $K_{m,n}$                   e)  $Q_n$

The **degree sequence** of a graph is the sequence of the degrees of the vertices of the graph in nonincreasing order. For example, the degree sequence of the graph  $G$  in Example 1 in this section is 4, 4, 4, 3, 2, 1, 0.

30. Find the degree sequences for each of the graphs in Exercises 21–25.

31. Find the degree sequence of each of the following graphs.

- a)  $K_4$                       b)  $C_4$                       c)  $W_4$
- d)  $K_{2,3}$                   e)  $Q_3$

32. What is the degree sequence of the bipartite graph  $K_{m,n}$  where  $m$  and  $n$  are positive integers? Explain your answer.

33. What is the degree sequence of  $K_n$ , where  $n$  is a positive integer? Explain your answer.

34. How many edges does a graph have if its degree sequence is 4, 3, 3, 2, 2? Draw such a graph.

35. How many edges does a graph have if its degree sequence is 5, 2, 2, 2, 2, 1? Draw such a graph.

A sequence  $d_1, d_2, \dots, d_n$  is called **graphic** if it is the degree sequence of a simple graph.

36. Determine whether each of these sequences is graphic. For those that are, draw a graph having the given degree sequence.

- a) 5, 4, 3, 2, 1, 0                      b) 6, 5, 4, 3, 2, 1
- c) 2, 2, 2, 2, 2, 2                      d) 3, 3, 3, 2, 2, 2
- e) 3, 3, 2, 2, 2, 2                      f) 1, 1, 1, 1, 1, 1
- g) 5, 3, 3, 3, 3, 3                      h) 5, 5, 4, 3, 2, 1

37. Determine whether each of these sequences is graphic. For those that are, draw a graph having the given degree sequence.

- a) 3, 3, 3, 3, 2                      b) 5, 4, 3, 2, 1
- c) 4, 4, 3, 2, 1                      d) 4, 4, 3, 3, 3
- e) 3, 2, 2, 1, 0                      f) 1, 1, 1, 1, 1

38. Suppose that  $d_1, d_2, \dots, d_n$  is a graphic sequence. Show that there is a simple graph with vertices  $v_1, v_2, \dots, v_n$  such that  $\deg(v_i) = d_i$  for  $i = 1, 2, \dots, n$  and  $v_1$  is adjacent to  $v_2, \dots, v_{d_1}$ .

\*39. Show that a sequence  $d_1, d_2, \dots, d_n$  of nonnegative integers in nonincreasing order is a graphic sequence if and only if the sequence obtained by reordering the terms of the sequence  $d_2 - 1, \dots, d_{d_1+1} - 1, d_{d_1+2}, \dots, d_n$  so that the terms are in nonincreasing order is a graphic sequence.

\*40. Use Exercise 39 to construct a recursive algorithm for determining whether a nonincreasing sequence of positive integers is graphic.

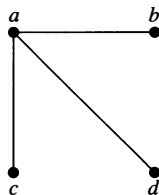
41. Show that every nonincreasing sequence of nonnegative integers with an even sum of its terms is the degree sequence of a pseudograph, that is, an undirected graph where loops are allowed. [Hint: Construct such a graph by first adding as many loops as possible at each vertex. Then add additional edges connecting vertices of odd degree. Explain why this construction works.]

42. How many subgraphs with at least one vertex does  $K_2$  have?

43. How many subgraphs with at least one vertex does  $K_3$  have?

44. How many subgraphs with at least one vertex does  $W_3$  have?

45. Draw all subgraphs of this graph.



46. Let  $G$  be a graph with  $v$  vertices and  $e$  edges. Let  $M$  be the maximum degree of the vertices of  $G$ , and let  $m$  be the minimum degree of the vertices of  $G$ . Show that

- a)  $2e/v \geq m$ .
- b)  $2e/v \leq M$ .

A simple graph is called **regular** if every vertex of this graph has the same degree. A regular graph is called  **$n$ -regular** if every vertex in this graph has degree  $n$ .

47. For which values of  $n$  are these graphs regular?

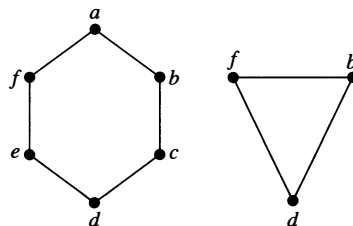
- a)  $K_n$                       b)  $C_n$                       c)  $W_n$                       d)  $Q_n$

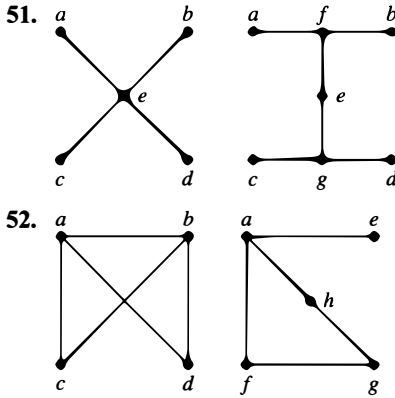
48. For which values of  $m$  and  $n$  is  $K_{m,n}$  regular?

49. How many vertices does a regular graph of degree four with 10 edges have?

In Exercises 50–52 find the union of the given pair of simple graphs. (Assume edges with the same endpoints are the same.)

50.





53. The **complementary graph**  $\overline{G}$  of a simple graph  $G$  has the same vertices as  $G$ . Two vertices are adjacent in  $\overline{G}$  if and only if they are not adjacent in  $G$ . Describe each of these graphs.

- a)  $\overline{K_n}$       b)  $\overline{K_{m,n}}$       c)  $\overline{C_n}$       d)  $\overline{Q_n}$

54. If  $G$  is a simple graph with 15 edges and  $\overline{G}$  has 13 edges, how many vertices does  $G$  have?
55. If the simple graph  $G$  has  $v$  vertices and  $e$  edges, how many edges does  $\overline{G}$  have?
56. If the degree sequence of the simple graph  $G$  is 4, 3, 3, 2, 2, what is the degree sequence of  $\overline{G}$ ?
57. If the degree sequence of the simple graph  $G$  is  $d_1, d_2, \dots, d_n$ , what is the degree sequence of  $\overline{G}$ ?
58. Show that if  $G$  is a bipartite simple graph with  $v$  vertices and  $e$  edges, then  $e \leq v^2/4$ .
59. Show that if  $G$  is a simple graph with  $n$  vertices, then the union of  $G$  and  $\overline{G}$  is  $K_n$ .

- \*60. Describe an algorithm to decide whether a graph is bipartite based on the fact that a graph is bipartite if and only if it is possible to color its vertices two different colors so that no two vertices of the same color are adjacent.

The **converse** of a directed graph  $G = (V, E)$ , denoted by  $G^{conv}$ , is the directed graph  $(V, F)$ , where the set  $F$  of edges of  $G^{conv}$  is obtained by reversing the direction of each edge in  $E$ .

61. Draw the converse of each of the graphs in Exercises 7–9 in Section 9.1.
62. Show that  $(G^{conv})^{conv} = G$  whenever  $G$  is a directed graph.
63. Show that the graph  $G$  is its own converse if and only if the relation associated with  $G$  (see Section 8.3) is symmetric.
64. Show that if a bipartite graph  $G = (V, E)$  is  $n$ -regular for some positive integer  $n$  (see the preamble to Exercise 47) and  $(V_1, V_2)$  is a bipartition of  $V$ , then  $|V_1| = |V_2|$ . That is, show that the two sets in a bipartition of the vertex set of an  $n$ -regular graph must contain the same number of vertices.
65. Draw the mesh network for interconnecting nine parallel processors.
66. In a variant of a mesh network for interconnecting  $n = m^2$  processors, processor  $P(i, j)$  is connected to the four processors  $P((i \pm 1) \bmod m, j)$  and  $P(i, (j \pm 1) \bmod m)$ , so that connections wrap around the edges of the mesh. Draw this variant of the mesh network for 16 processors.
67. Show that every pair of processors in a mesh network of  $n = m^2$  processors can communicate using  $O(\sqrt{n}) = O(m)$  hops between directly connected processors.

## 9.3 Representing Graphs and Graph Isomorphism

### Introduction

There are many useful ways to represent graphs. As we will see throughout this chapter, in working with a graph it is helpful to be able to choose its most convenient representation. In this section we will show how to represent graphs in several different ways.

Sometimes, two graphs have exactly the same form, in the sense that there is a one-to-one correspondence between their vertex sets that preserves edges. In such a case, we say that the two graphs are **isomorphic**. Determining whether two graphs are isomorphic is an important problem of graph theory that we will study in this section.

### Representing Graphs

One way to represent a graph without multiple edges is to list all the edges of this graph. Another way to represent a graph with no multiple edges is to use **adjacency lists**, which specify the vertices that are adjacent to each vertex of the graph.

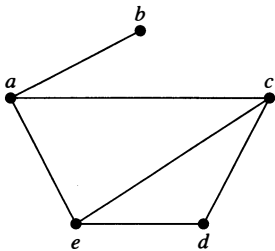


FIGURE 1 A Simple Graph.

TABLE 1 An Adjacency List for a Simple Graph.	
Vertex	Adjacent Vertices
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

**EXAMPLE 1** Use adjacency lists to describe the simple graph given in Figure 1.

*Solution:* Table 1 lists those vertices adjacent to each of the vertices of the graph.

**EXAMPLE 2** Represent the directed graph shown in Figure 2 by listing all the vertices that are the terminal vertices of edges starting at each vertex of the graph.

*Solution:* Table 2 represents the directed graph shown in Figure 2.

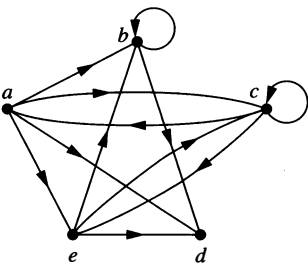


FIGURE 2 A Directed Graph.

TABLE 2 An Adjacency List for a Directed Graph.	
Initial Vertex	Terminal Vertices
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

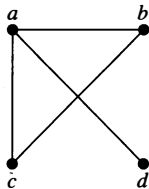


FIGURE 3 Simple Graph.



### Adjacency Matrices

Carrying out graph algorithms using the representation of graphs by lists of edges, or by adjacency lists, can be cumbersome if there are many edges in the graph. To simplify computation, graphs can be represented using matrices. Two types of matrices commonly used to represent graphs will be presented here. One is based on the adjacency of vertices, and the other is based on incidence of vertices and edges.

Suppose that  $G = (V, E)$  is a simple graph where  $|V| = n$ . Suppose that the vertices of  $G$  are listed arbitrarily as  $v_1, v_2, \dots, v_n$ . The **adjacency matrix**  $A$  (or  $A_G$ ) of  $G$ , with respect to this listing of the vertices, is the  $n \times n$  zero-one matrix with 1 as its  $(i, j)$ th entry when  $v_i$  and  $v_j$  are adjacent, and 0 as its  $(i, j)$ th entry when they are not adjacent. In other words, if its adjacency matrix is  $A = [a_{ij}]$ , then

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

**EXAMPLE 3** Use an adjacency matrix to represent the graph shown in Figure 3.

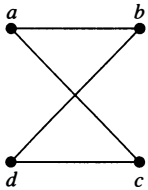


*Solution:* We order the vertices as  $a, b, c, d$ . The matrix representing this graph is

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

**EXAMPLE 4** Draw a graph with the adjacency matrix

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$



**FIGURE 4**  
A Graph with the  
Given Adjacency  
Matrix.

with respect to the ordering of vertices  $a, b, c, d$ .

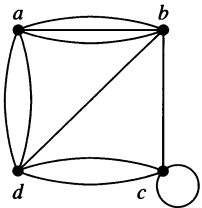
*Solution:* A graph with this adjacency matrix is shown in Figure 4.

Note that an adjacency matrix of a graph is based on the ordering chosen for the vertices. Hence, there are as many as  $n!$  different adjacency matrices for a graph with  $n$  vertices, because there are  $n!$  different orderings of  $n$  vertices.

The adjacency matrix of a simple graph is symmetric, that is,  $a_{ij} = a_{ji}$ , because both of these entries are 1 when  $v_i$  and  $v_j$  are adjacent, and both are 0 otherwise. Furthermore, because a simple graph has no loops, each entry  $a_{ii}$ ,  $i = 1, 2, 3, \dots, n$ , is 0.

Adjacency matrices can also be used to represent undirected graphs with loops and with multiple edges. A loop at the vertex  $a_i$  is represented by a 1 at the  $(i, i)$ th position of the adjacency matrix. When multiple edges are present, the adjacency matrix is no longer a zero–one matrix, because the  $(i, j)$ th entry of this matrix equals the number of edges that are associated to  $\{a_i, a_j\}$ . All undirected graphs, including multigraphs and pseudographs, have symmetric adjacency matrices.

**EXAMPLE 5** Use an adjacency matrix to represent the pseudograph shown in Figure 5.



**FIGURE 5**  
A Pseudograph.

*Solution:* The adjacency matrix using the ordering of vertices  $a, b, c, d$  is

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}.$$

We used zero–one matrices in Chapter 8 to represent directed graphs. The matrix for a directed graph  $G = (V, E)$  has a 1 in its  $(i, j)$ th position if there is an edge from  $v_i$  to  $v_j$ , where  $v_1, v_2, \dots, v_n$  is an arbitrary listing of the vertices of the directed graph. In other words, if  $A = [a_{ij}]$  is the adjacency matrix for the directed graph with respect to this listing of the vertices, then

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

The adjacency matrix for a directed graph does not have to be symmetric, because there may not be an edge from  $a_j$  to  $a_i$  when there is an edge from  $a_i$  to  $a_j$ .

Adjacency matrices can also be used to represent directed multigraphs. Again, such matrices are not zero–one matrices when there are multiple edges in the same direction connecting two vertices. In the adjacency matrix for a directed multigraph,  $a_{ij}$  equals the number of edges that are associated to  $(v_i, v_j)$ .

**TRADE-OFFS BETWEEN ADJACENCY LISTS AND ADJACENCY MATRICES** When a simple graph contains relatively few edges, that is, when it is **sparse**, it is usually preferable to use adjacency lists rather than an adjacency matrix to represent the graph. For example, if each vertex has degree not exceeding  $c$ , where  $c$  is a constant much smaller than  $n$ , then each adjacency list contains  $c$  or fewer vertices. Hence, there are no more than  $cn$  items in all these adjacency lists. On the other hand, the adjacency matrix for the graph has  $n^2$  entries. Note, however, that the adjacency matrix of a sparse graph is a **sparse matrix**, that is, a matrix with few nonzero entries, and there are special techniques for representing, and computing with, sparse matrices.

Now suppose that a simple graph is **dense**, that is, suppose that it contains many edges, such as a graph that contains more than half of all possible edges. In this case, using an adjacency matrix to represent the graph is usually preferable over using adjacency lists. To see why, we compare the complexity of determining whether the possible edge  $\{v_i, v_j\}$  is present. Using an adjacency matrix, we can determine whether this edge is present by examining the  $(i, j)$ th entry in the matrix. This entry is 1 if the graph contains this edge and is 0 otherwise. Consequently, we need make only one comparison, namely, comparing this entry with 0, to determine whether this edge is present. On the other hand, when we use adjacency lists to represent the graph, we need to search the list of vertices adjacent to either  $v_i$  or  $v_j$  to determine whether this edge is present. This can require  $\Theta(|V|)$  comparisons when many edges are present.

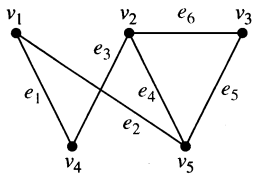
## Incidence Matrices

Another common way to represent graphs is to use **incidence matrices**. Let  $G = (V, E)$  be an undirected graph. Suppose that  $v_1, v_2, \dots, v_n$  are the vertices and  $e_1, e_2, \dots, e_m$  are the edges of  $G$ . Then the incidence matrix with respect to this ordering of  $V$  and  $E$  is the  $n \times m$  matrix  $\mathbf{M} = [m_{ij}]$ , where

$$m_{ij} = \begin{cases} 1 & \text{when edge } e_j \text{ is incident with } v_i, \\ 0 & \text{otherwise.} \end{cases}$$

**EXAMPLE 6** Represent the graph shown in Figure 6 with an incidence matrix.

*Solution:* The incidence matrix is

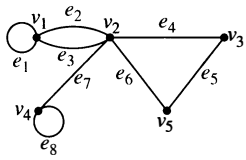


$$\begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix}.$$

**FIGURE 6** An Undirected Graph.

Incidence matrices can also be used to represent multiple edges and loops. Multiple edges are represented in the incidence matrix using columns with identical entries, because these edges are incident with the same pair of vertices. Loops are represented using a column with exactly one entry equal to 1, corresponding to the vertex that is incident with this loop.

**EXAMPLE 7** Represent the pseudograph shown in Figure 7 using an incidence matrix.



**FIGURE 7**  
**A Pseudograph.**

*Solution:* The incidence matrix for this graph is

$$\begin{array}{c} \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

## Isomorphism of Graphs

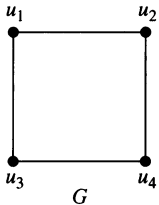
We often need to know whether it is possible to draw two graphs in the same way. For instance, in chemistry, graphs are used to model compounds. Different compounds can have the same molecular formula but can differ in structure. Such compounds will be represented by graphs that cannot be drawn in the same way. The graphs representing previously known compounds can be used to determine whether a supposedly new compound has been studied before.

There is a useful terminology for graphs with the same structure.

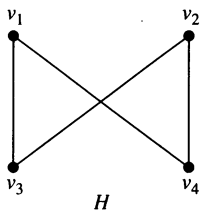
**DEFINITION 1** The simple graphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  are *isomorphic* if there is a one-to-one and onto function  $f$  from  $V_1$  to  $V_2$  with the property that  $a$  and  $b$  are adjacent in  $G_1$  if and only if  $f(a)$  and  $f(b)$  are adjacent in  $G_2$ , for all  $a$  and  $b$  in  $V_1$ . Such a function  $f$  is called an *isomorphism*.\*

In other words, when two simple graphs are isomorphic, there is a one-to-one correspondence between vertices of the two graphs that preserves the adjacency relationship. Isomorphism of simple graphs is an equivalence relation. (We leave the verification of this as Exercise 45 at the end of this section.)

**EXAMPLE 8** Show that the graphs  $G = (V, E)$  and  $H = (W, F)$ , displayed in Figure 8, are isomorphic.



*Solution:* The function  $f$  with  $f(u_1) = v_1$ ,  $f(u_2) = v_4$ ,  $f(u_3) = v_3$ , and  $f(u_4) = v_2$  is a one-to-one correspondence between  $V$  and  $W$ . To see that this correspondence preserves adjacency, note that adjacent vertices in  $G$  are  $u_1$  and  $u_2$ ,  $u_1$  and  $u_3$ ,  $u_2$  and  $u_4$ , and  $u_3$  and  $u_4$ , and each of the pairs  $f(u_1) = v_1$  and  $f(u_2) = v_4$ ,  $f(u_1) = v_1$  and  $f(u_3) = v_3$ ,  $f(u_2) = v_4$  and  $f(u_4) = v_2$ , and  $f(u_3) = v_3$  and  $f(u_4) = v_2$  are adjacent in  $H$ .



**FIGURE 8** The  
Graphs  $G$  and  $H$ .

It is often difficult to determine whether two simple graphs are isomorphic. There are  $n!$  possible one-to-one correspondences between the vertex sets of two simple graphs with  $n$  vertices. Testing each such correspondence to see whether it preserves adjacency and nonadjacency is impractical if  $n$  is at all large.

Sometimes it is not hard to show that two graphs are not isomorphic. In particular, we can show that two graphs are not isomorphic if we can find a property only one of the two graphs has, but that is preserved by isomorphism. A property preserved by isomorphism of graphs is called a **graph invariant**. For instance, isomorphic simple graphs must have the same number of vertices, because there is a one-to-one correspondence between the sets of vertices of the graphs.

\*The word *isomorphism* comes from the Greek roots *isos* for “equal” and *morphe* for “form.”

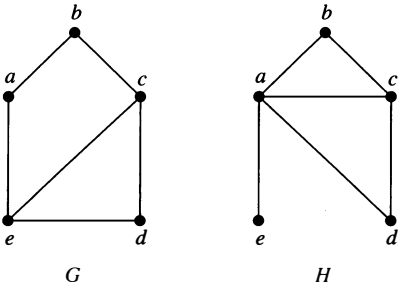


FIGURE 9 The Graphs  $G$  and  $H$ .

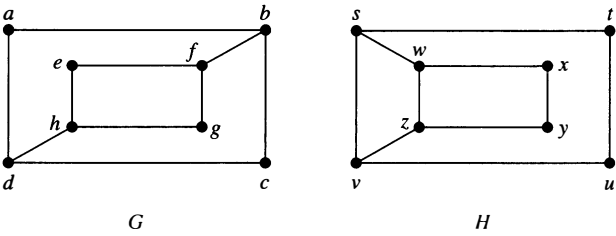


FIGURE 10 The Graphs  $G$  and  $H$ .



Isomorphic simple graphs also must have the same number of edges, because the one-to-one correspondence between vertices establishes a one-to-one correspondence between edges. In addition, the degrees of the vertices in isomorphic simple graphs must be the same. That is, a vertex  $v$  of degree  $d$  in  $G$  must correspond to a vertex  $f(v)$  of degree  $d$  in  $H$ , because a vertex  $w$  in  $G$  is adjacent to  $v$  if and only if  $f(v)$  and  $f(w)$  are adjacent in  $H$ .

**EXAMPLE 9** Show that graphs displayed in Figure 9 are not isomorphic.



**Solution:** Both  $G$  and  $H$  have five vertices and six edges. However,  $H$  has a vertex of degree one, namely,  $e$ , whereas  $G$  has no vertices of degree one. It follows that  $G$  and  $H$  are not isomorphic.

The number of vertices, the number of edges, and the number of vertices of each degree are all invariants under isomorphism. If any of these quantities differ in two simple graphs, these graphs cannot be isomorphic. However, when these invariants are the same, it does not necessarily mean that the two graphs are isomorphic. There are no useful sets of invariants currently known that can be used to determine whether simple graphs are isomorphic.

**EXAMPLE 10** Determine whether the graphs shown in Figure 10 are isomorphic.

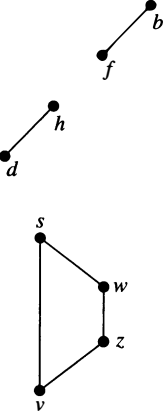


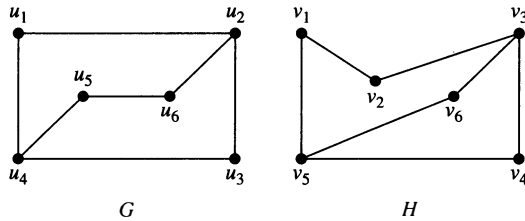
FIGURE 11 The Subgraphs of  $G$  and  $H$  Made Up of Vertices of Degree Three and the Edges Connecting Them.

**Solution:** The graphs  $G$  and  $H$  both have eight vertices and 10 edges. They also both have four vertices of degree two and four of degree three. Because these invariants all agree, it is still conceivable that these graphs are isomorphic.

However,  $G$  and  $H$  are not isomorphic. To see this, note that because  $\deg(a) = 2$  in  $G$ ,  $a$  must correspond to either  $t$ ,  $u$ ,  $x$ , or  $y$  in  $H$ , because these are the vertices of degree two in  $H$ . However, each of these four vertices in  $H$  is adjacent to another vertex of degree two in  $H$ , which is not true for  $a$  in  $G$ .

Another way to see that  $G$  and  $H$  are not isomorphic is to note that the subgraphs of  $G$  and  $H$  made up of vertices of degree three and the edges connecting them must be isomorphic if these two graphs are isomorphic (the reader should verify this). However, these subgraphs, shown in Figure 11, are not isomorphic.

To show that a function  $f$  from the vertex set of a graph  $G$  to the vertex set of a graph  $H$  is an isomorphism, we need to show that  $f$  preserves the presence and absence of edges. One helpful way to do this is to use adjacency matrices. In particular, to show that  $f$  is an isomorphism, we can show that the adjacency matrix of  $G$  is the same as the adjacency matrix of  $H$ , when rows and columns are labeled to correspond to the images under  $f$  of the vertices in  $G$  that are the labels of these rows and columns in the adjacency matrix of  $G$ . We illustrate how this is done in Example 11.

FIGURE 12 Graphs  $G$  and  $H$ .

**EXAMPLE 11** Determine whether the graphs  $G$  and  $H$  displayed in Figure 12 are isomorphic.

*Solution:* Both  $G$  and  $H$  have six vertices and seven edges. Both have four vertices of degree two and two vertices of degree three. It is also easy to see that the subgraphs of  $G$  and  $H$  consisting of all vertices of degree two and the edges connecting them are isomorphic (as the reader should verify). Because  $G$  and  $H$  agree with respect to these invariants, it is reasonable to try to find an isomorphism  $f$ .

We now will define a function  $f$  and then determine whether it is an isomorphism. Because  $\deg(u_1) = 2$  and because  $u_1$  is not adjacent to any other vertex of degree two, the image of  $u_1$  must be either  $v_4$  or  $v_6$ , the only vertices of degree two in  $H$  not adjacent to a vertex of degree two. We arbitrarily set  $f(u_1) = v_6$ . [If we found that this choice did not lead to isomorphism, we would then try  $f(u_1) = v_4$ .] Because  $u_2$  is adjacent to  $u_1$ , the possible images of  $u_2$  are  $v_3$  and  $v_5$ . We arbitrarily set  $f(u_2) = v_3$ . Continuing in this way, using adjacency of vertices and degrees as a guide, we set  $f(u_3) = v_4$ ,  $f(u_4) = v_5$ ,  $f(u_5) = v_1$ , and  $f(u_6) = v_2$ . We now have a one-to-one correspondence between the vertex set of  $G$  and the vertex set of  $H$ , namely,  $f(u_1) = v_6$ ,  $f(u_2) = v_3$ ,  $f(u_3) = v_4$ ,  $f(u_4) = v_5$ ,  $f(u_5) = v_1$ ,  $f(u_6) = v_2$ . To see whether  $f$  preserves edges, we examine the adjacency matrix of  $G$ ,

$$\mathbf{A}_G = \begin{matrix} & \begin{matrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{matrix} \\ \begin{matrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix},$$

and the adjacency matrix of  $H$  with the rows and columns labeled by the images of the corresponding vertices in  $G$ ,

$$\mathbf{A}_H = \begin{matrix} & \begin{matrix} v_6 & v_3 & v_4 & v_5 & v_1 & v_2 \end{matrix} \\ \begin{matrix} v_6 \\ v_3 \\ v_4 \\ v_5 \\ v_1 \\ v_2 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}.$$

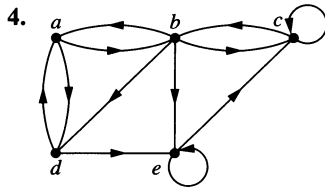
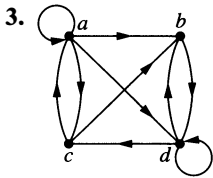
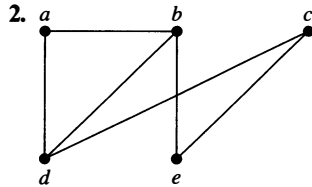
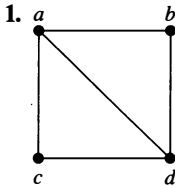
Because  $A_G = A_H$ , it follows that  $f$  preserves edges. We conclude that  $f$  is an isomorphism, so  $G$  and  $H$  are isomorphic. Note that if  $f$  turned out not to be an isomorphism, we would *not* have established that  $G$  and  $H$  are not isomorphic, because another correspondence of the vertices in  $G$  and  $H$  may be an isomorphism. ◀



The best algorithms known for determining whether two graphs are isomorphic have exponential worst-case time complexity (in the number of vertices of the graphs). However, linear average-case time complexity algorithms are known that solve this problem, and there is some hope that an algorithm with polynomial worst-case time complexity for determining whether two graphs are isomorphic can be found. The best practical algorithm, called NAUTY, can be used to determine whether two graphs with as many as 100 vertices are isomorphic in less than 1 second on a modern PC. The software for NAUTY can be downloaded over the Internet and experimented with.

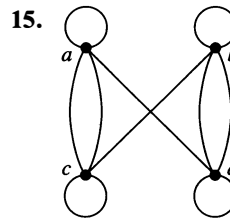
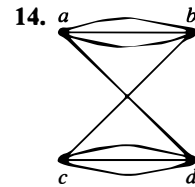
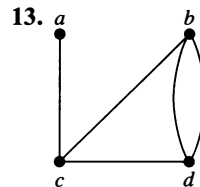
## Exercises

In Exercises 1–4 use an adjacency list to represent the given graph.



12. 
$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

In Exercises 13–15 represent the given graph using an adjacency matrix.



5. Represent the graph in Exercise 1 with an adjacency matrix.
6. Represent the graph in Exercise 2 with an adjacency matrix.
7. Represent the graph in Exercise 3 with an adjacency matrix.
8. Represent the graph in Exercise 4 with an adjacency matrix.
9. Represent each of these graphs with an adjacency matrix.

- a)  $K_4$       b)  $K_{1,4}$       c)  $K_{2,3}$   
 d)  $C_4$       e)  $W_4$       f)  $Q_3$

In Exercises 10–12 draw a graph with the given adjacency matrix.

10. 
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

11. 
$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

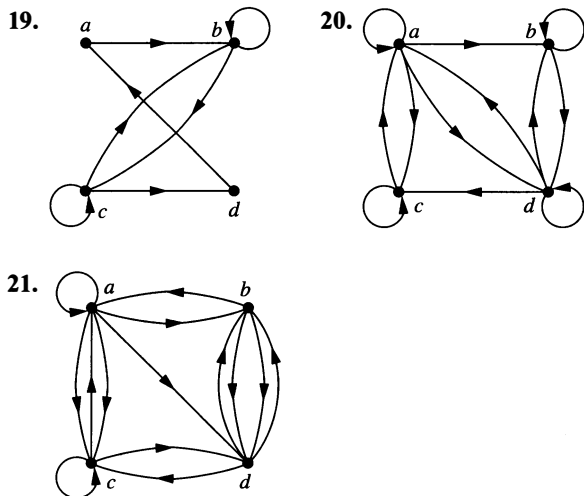
In Exercises 16–18 draw an undirected graph represented by the given adjacency matrix.

16. 
$$\begin{bmatrix} 1 & 3 & 2 \\ 3 & 0 & 4 \\ 2 & 4 & 0 \end{bmatrix}$$

17. 
$$\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

18. 
$$\begin{bmatrix} 0 & 1 & 3 & 0 & 4 \\ 1 & 2 & 1 & 3 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ 0 & 3 & 0 & 0 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$$

In Exercises 19–21 find the adjacency matrix of the given directed multigraph.



In Exercises 22–24 draw the graph represented by the given adjacency matrix.

22.  $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$     23.  $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix}$     24.  $\begin{bmatrix} 0 & 2 & 3 & 0 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 1 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}$

25. Is every zero-one square matrix that is symmetric and has zeros on the diagonal the adjacency matrix of a simple graph?

26. Use an incidence matrix to represent the graphs in Exercises 1 and 2.

27. Use an incidence matrix to represent the graphs in Exercises 13–15.

\*28. What is the sum of the entries in a row of the adjacency matrix for an undirected graph? For a directed graph?

\*29. What is the sum of the entries in a column of the adjacency matrix for an undirected graph? For a directed graph?

30. What is the sum of the entries in a row of the incidence matrix for an undirected graph?

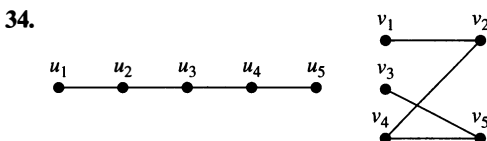
31. What is the sum of the entries in a column of the incidence matrix for an undirected graph?

\*32. Find an adjacency matrix for each of these graphs.

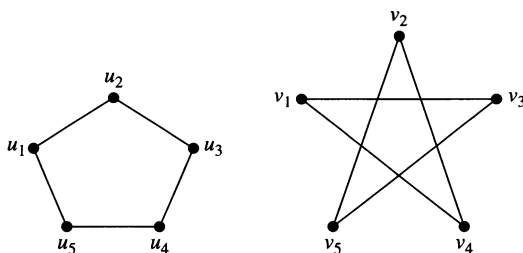
a)  $K_n$     b)  $C_n$     c)  $W_n$     d)  $K_{m,n}$     e)  $Q_n$

\*33. Find incidence matrices for the graphs in parts (a)–(d) of Exercise 32.

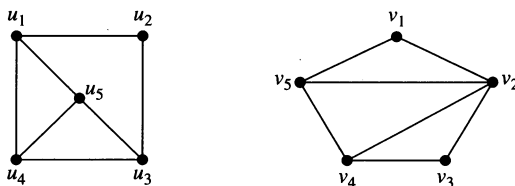
In Exercises 34–44 determine whether the given pair of graphs is isomorphic. Exhibit an isomorphism or provide a rigorous argument that none exists.



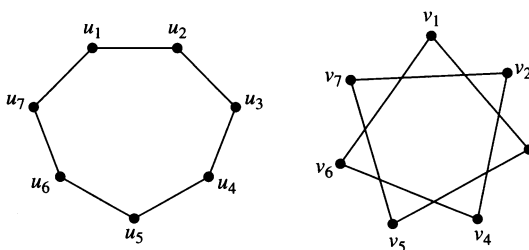
35.



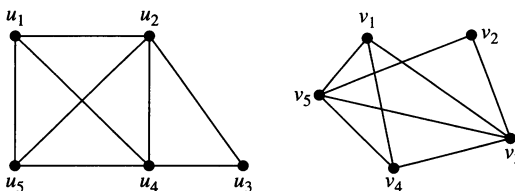
36.



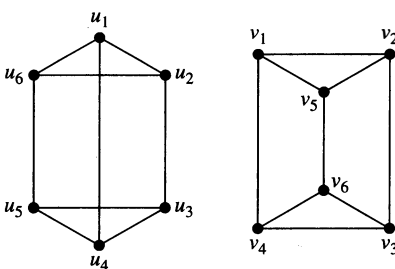
37.



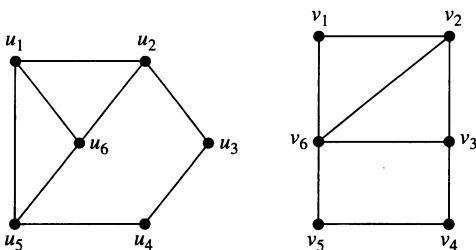
38.

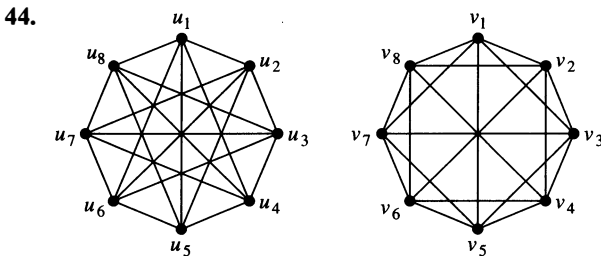
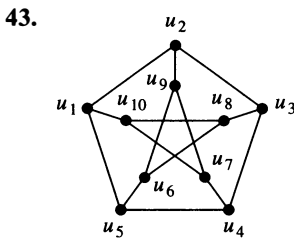
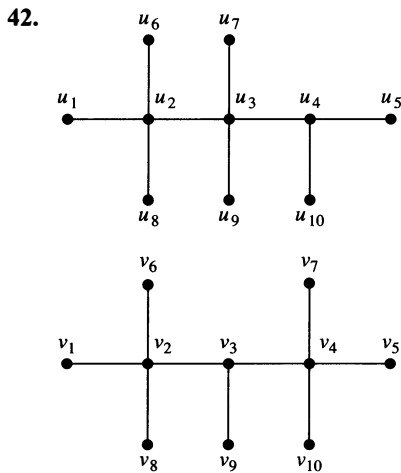
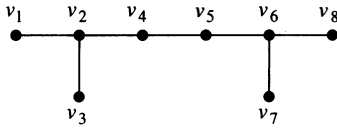
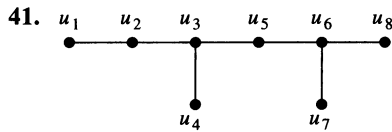


39.



40.





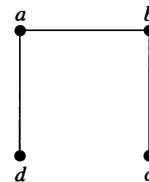
47. Describe the row and column of an adjacency matrix of a graph corresponding to an isolated vertex.
48. Describe the row of an incidence matrix of a graph corresponding to an isolated vertex.
49. Show that the vertices of a bipartite graph with two or more vertices can be ordered so that its adjacency matrix has the form

$$\begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{B} & \mathbf{0} \end{bmatrix},$$

where the four entries shown are rectangular blocks.

A simple graph  $G$  is called **self-complementary** if  $G$  and  $\overline{G}$  are isomorphic.

50. Show that this graph is self-complementary.



51. Find a self-complementary simple graph with five vertices.
- \*52. Show that if  $G$  is a self-complementary simple graph with  $v$  vertices, then  $v \equiv 0$  or  $1 \pmod{4}$ .
53. For which integers  $n$  is  $C_n$  self-complementary?
54. How many nonisomorphic simple graphs are there with  $n$  vertices, when  $n$  is
- a) 2?                      b) 3?                      c) 4?
55. How many nonisomorphic simple graphs are there with five vertices and three edges?
56. How many nonisomorphic simple graphs are there with six vertices and four edges?
57. Are the simple graphs with the following adjacency matrices isomorphic?

a)  $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

b)  $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

c)  $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

45. Show that isomorphism of simple graphs is an equivalence relation.
46. Suppose that  $G$  and  $H$  are isomorphic simple graphs. Show that their complementary graphs  $\overline{G}$  and  $\overline{H}$  are also isomorphic.



58. Determine whether the graphs without loops with these incidence matrices are isomorphic.

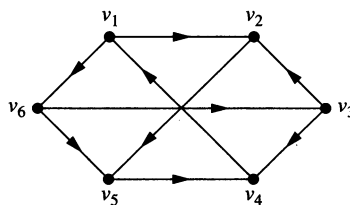
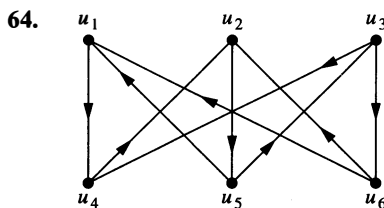
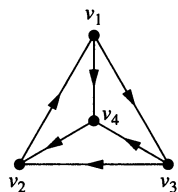
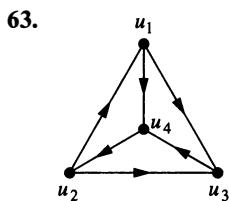
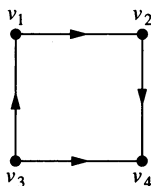
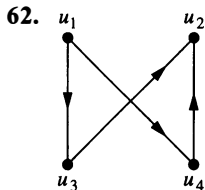
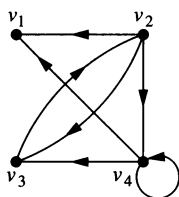
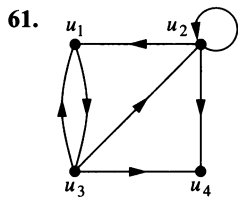
a)  $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

b)  $\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$

59. Extend the definition of isomorphism of simple graphs to undirected graphs containing loops and multiple edges.

60. Define isomorphism of directed graphs.

In Exercises 61–64 determine whether the given pair of directed graphs are isomorphic. (See Exercise 60.)



65. Show that if  $G$  and  $H$  are isomorphic directed graphs, then the converses of  $G$  and  $H$  (defined in the preamble of Exercise 61 of Section 9.2) are also isomorphic.

66. Show that the property that a graph is bipartite is an isomorphic invariant.

67. Find a pair of nonisomorphic graphs with the same degree sequence such that one graph is bipartite, but the other graph is not bipartite.

\*68. How many nonisomorphic directed simple graphs are there with  $n$  vertices, when  $n$  is

- a) 2?                      b) 3?                      c) 4?

\*69. What is the product of the incidence matrix and its transpose for an undirected graph?

\*70. How much storage is needed to represent a simple graph with  $v$  vertices and  $e$  edges using

- a) adjacency lists?  
b) an adjacency matrix?  
c) an incidence matrix?

A **devil's pair** for a purported isomorphism test is a pair of nonisomorphic graphs that the test fails to show are not isomorphic.

71. Find a devil's pair for the test that checks the degree sequence (defined in the preamble to Exercise 30 in Section 9.2) in two graphs to make sure they agree.

## 9.4 Connectivity

### Introduction

Many problems can be modeled with paths formed by traveling along the edges of graphs. For instance, the problem of determining whether a message can be sent between two computers using intermediate links can be studied with a graph model. Problems of efficiently planning routes for mail delivery, garbage pickup, diagnostics in computer networks, and so on can be solved using models that involve paths in graphs.

## Paths

Informally, a **path** is a sequence of edges that begins at a vertex of a graph and travels from vertex to vertex along edges of the graph.

A formal definition of paths and related terminology is given in Definition 1.

**DEFINITION 1** Let  $n$  be a nonnegative integer and  $G$  an undirected graph. A *path* of length  $n$  from  $u$  to  $v$  in  $G$  is a sequence of  $n$  edges  $e_1, \dots, e_n$  of  $G$  such that  $e_1$  is associated with  $\{x_0, x_1\}$ ,  $e_2$  is associated with  $\{x_1, x_2\}$ , and so on, with  $e_n$  associated with  $\{x_{n-1}, x_n\}$ , where  $x_0 = u$  and  $x_n = v$ . When the graph is simple, we denote this path by its vertex sequence  $x_0, x_1, \dots, x_n$  (because listing these vertices uniquely determines the path). The path is a *circuit* if it begins and ends at the same vertex, that is, if  $u = v$ , and has length greater than zero. The path or circuit is said to *pass through* the vertices  $x_1, x_2, \dots, x_{n-1}$  or *traverse* the edges  $e_1, e_2, \dots, e_n$ . A path or circuit is *simple* if it does not contain the same edge more than once.

When it is not necessary to distinguish between multiple edges, we will denote a path  $e_1, e_2, \dots, e_n$ , where  $e_i$  is associated with  $\{x_{i-1}, x_i\}$  for  $i = 1, 2, \dots, n$  by its vertex sequence  $x_0, x_1, \dots, x_n$ . This notation identifies a path only up to the vertices it passes through. There may be more than one path that passes through this sequence of vertices. Note that a path of length zero consists of a single vertex.

**Remark:** There is considerable variation of terminology concerning the concepts defined in Definition 1. For instance, in some books, the term **walk** is used instead of *path*, where a walk is defined to be an alternating sequence of vertices and edges of a graph,  $v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$ , where  $v_{i-1}$  and  $v_i$  are the endpoints of  $e_i$  for  $i = 1, 2, \dots, n$ . When this terminology is used, **closed walk** is used instead of *circuit* to indicate a walk that begins and ends at the same vertex, and **trail** is used to denote a walk that has no repeated edge (replacing the term *simple path*). When this terminology is used, the terminology **path** is often used for a trail with no repeated vertices, conflicting with the terminology in Definition 1. Because of this variation in terminology, you will need to make sure which set of definitions are used in a particular book or article when you read about traversing edges of a graph. The text [GrYe06] is a good reference for the alternative terminology described in this remark.

**EXAMPLE 1** In the simple graph shown in Figure 1,  $a, d, c, f, e$  is a simple path of length 4, because  $\{a, d\}$ ,  $\{d, c\}$ ,  $\{c, f\}$ , and  $\{f, e\}$  are all edges. However,  $d, e, c, a$  is not a path, because  $\{e, c\}$  is not an edge. Note that  $b, c, f, e, b$  is a circuit of length 4 because  $\{b, c\}$ ,  $\{c, f\}$ ,  $\{f, e\}$ , and  $\{e, b\}$  are edges, and this path begins and ends at  $b$ . The path  $a, b, e, d, a, b$ , which is of length 5, is not simple because it contains the edge  $\{a, b\}$  twice. ◀

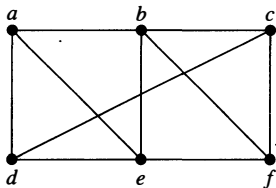


FIGURE 1 A Simple Graph.

Paths and circuits in directed graphs were introduced in Chapter 8. We now provide more general definitions.

**DEFINITION 2** Let  $n$  be a nonnegative integer and  $G$  a directed graph. A *path* of length  $n$  from  $u$  to  $v$  in  $G$  is a sequence of edges  $e_1, e_2, \dots, e_n$  of  $G$  such that  $e_1$  is associated with  $(x_0, x_1)$ ,  $e_2$  is associated with  $(x_1, x_2)$ , and so on, with  $e_n$  associated with  $(x_{n-1}, x_n)$ , where  $x_0 = u$  and  $x_n = v$ . When there are no multiple edges in the directed graph, this path is denoted by its vertex sequence  $x_0, x_1, x_2, \dots, x_n$ . A path of length greater than zero that begins and ends at the same vertex is called a *circuit* or *cycle*. A path or circuit is called *simple* if it does not contain the same edge more than once.

**Remark:** Terminology other than that given in Definition 2 is often used for the concepts defined there. In particular, the alternative terminology that uses *walk*, *closed walk*, *trail*, and *path* (described in the remarks following Definition 1) may be used for directed graphs. See [GrYe06] for details.

Note that the terminal vertex of an edge in a path is the initial vertex of the next edge in the path. When it is not necessary to distinguish between multiple edges, we will denote a path  $e_1, e_2, \dots, e_n$ , where  $e_i$  is associated with  $(x_{i-1}, x_i)$  for  $i = 1, 2, \dots, n$  by its vertex sequence  $x_0, x_1, \dots, x_n$ . The notation identifies a path only up to the vertices it passes through. There may be more than one path that passes through this sequence of vertices.

Paths represent useful information in many graph models, as Examples 2–4 demonstrate.

**EXAMPLE 2 Paths in Acquaintanceship Graphs** In an acquaintanceship graph there is a path between two people if there is a chain of people linking these people, where two people adjacent in the chain know one another. For example, in Figure 7 in Section 9.1, there is a chain of six people linking Kamini and Ching. Many social scientists have conjectured that almost every pair of people in the world are linked by a small chain of people, perhaps containing just five or fewer people. This would mean that almost every pair of vertices in the acquaintanceship graph containing all people in the world is linked by a path of length not exceeding four. The play *Six Degrees of Separation* by John Guare is based on this notion. ◀



**EXAMPLE 3 Paths in Collaboration Graphs** In a collaboration graph two vertices  $a$  and  $b$ , which represent authors, are connected by a path when there is a sequence of authors beginning at  $a$  and ending at  $b$  such that the two authors represented by the endpoints of each edge have written a joint paper. In the collaboration graph of all mathematicians, the **Erdős number** of a mathematician  $m$  (defined in terms of relations in Supplementary Exercise 14 in Chapter 8) is the length of the shortest path between  $m$  and the vertex representing the extremely prolific mathematician Paul Erdős (who died in 1996). That is, the Erdős number of a mathematician is the length of the shortest chain of mathematicians that begins with Paul Erdős and ends with this mathematician, where each adjacent pair of mathematicians have written a joint paper. The number of mathematicians with each Erdős number as of early 2006, according to the Erdős Number Project, is shown in Table 1. ◀



**EXAMPLE 4 Paths in the Hollywood Graph** In the Hollywood graph (see Example 4 in Section 9.1) two vertices  $a$  and  $b$  are linked when there is a chain of actors linking  $a$  and  $b$ , where every two actors adjacent in the chain have acted in the same movie. In the Hollywood graph, the **Bacon number** of an actor  $c$  is defined to be the length of the shortest path connecting  $c$  and the well-known actor Kevin Bacon. As new movies are made, including new ones with Kevin Bacon, the Bacon



<b>TABLE 1 The Number of Mathematicians with a Given Erdős Number (as of early 2006).</b>	
<i>Erdős Number</i>	<i>Number of People</i>
0	1
1	504
2	6,593
3	33,605
4	83,642
5	87,760
6	40,014
7	11,591
8	3,146
9	819
10	244
11	68
12	23
13	5

<b>TABLE 2 The Number of Actors with a Given Bacon Number (as of early 2006).</b>	
<i>Bacon Number</i>	<i>Number of People</i>
0	1
1	1,902
2	160,463
3	457,231
4	111,310
5	8,168
6	810
7	81
8	14

number of actors can change. In Table 2 we show the number of actors with each Bacon number as of early 2006 using data from the Oracle of Bacon website. ◀

## Connectedness In Undirected Graphs

When does a computer network have the property that every pair of computers can share information, if messages can be sent through one or more intermediate computers? When a graph is used to represent this computer network, where vertices represent the computers and edges represent the communication links, this question becomes: When is there always a path between two vertices in the graph?

**DEFINITION 3** An undirected graph is called *connected* if there is a path between every pair of distinct vertices of the graph.

Thus, any two computers in the network can communicate if and only if the graph of this network is connected.

**EXAMPLE 5** The graph  $G_1$  in Figure 2 is connected, because for every pair of distinct vertices there is a path between them (the reader should verify this). However, the graph  $G_2$  in Figure 2 is not connected. For instance, there is no path in  $G_2$  between vertices  $a$  and  $d$ . ◀

We will need the following theorem in Chapter 10.

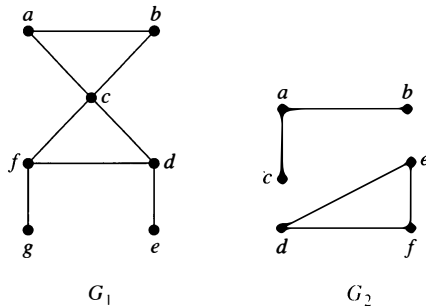


FIGURE 2 The Graphs  $G_1$  and  $G_2$ .

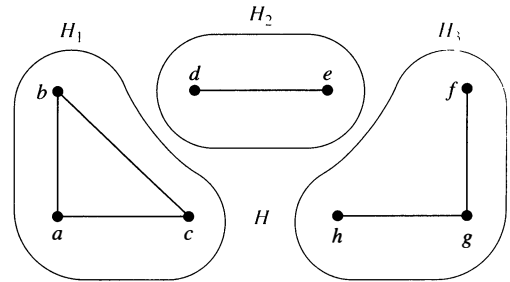


FIGURE 3 The Graph  $H$  and Its Connected Components  $H_1$ ,  $H_2$ , and  $H_3$ .

**THEOREM 1** There is a simple path between every pair of distinct vertices of a connected undirected graph.

*Proof:* Let  $u$  and  $v$  be two distinct vertices of the connected undirected graph  $G = (V, E)$ . Because  $G$  is connected, there is at least one path between  $u$  and  $v$ . Let  $x_0, x_1, \dots, x_n$ , where  $x_0 = u$  and  $x_n = v$ , be the vertex sequence of a path of least length. This path of least length is simple. To see this, suppose it is not simple. Then  $x_i = x_j$  for some  $i$  and  $j$  with  $0 \leq i < j$ . This means that there is a path from  $u$  to  $v$  of shorter length with vertex sequence  $x_0, x_1, \dots, x_{i-1}, x_j, \dots, x_n$  obtained by deleting the edges corresponding to the vertex sequence  $x_i, \dots, x_{j-1}$ .  $\blacktriangleleft$

A **connected component** of a graph  $G$  is a connected subgraph of  $G$  that is not a proper subgraph of another connected subgraph of  $G$ . That is, a connected component of a graph  $G$  is a maximal connected subgraph of  $G$ . A graph  $G$  that is not connected has two or more connected components that are disjoint and have  $G$  as their union.



**EXAMPLE 6** What are the connected components of the graph  $H$  shown in Figure 3?

*Solution:* The graph  $H$  is the union of three disjoint connected subgraphs  $H_1$ ,  $H_2$ , and  $H_3$ , shown in Figure 3. These three subgraphs are the connected components of  $H$ .  $\blacktriangleleft$

**EXAMPLE 7 Connected Components of Call Graphs** Two vertices  $x$  and  $y$  are in the same component of a telephone call graph (see Example 7 in Section 9.1) when there is a sequence of telephone calls beginning at  $x$  and ending at  $y$ . When a call graph for telephone calls made during a particular day in the AT&T network was analyzed, this graph was found to have 53,767,087 vertices, more than 170 million edges, and more than 3.7 million connected components. Most of these components were small; approximately three-fourths consisted of two vertices representing pairs of telephone numbers that called only each other. This graph has one huge connected component with 44,989,297 vertices comprising more than 80% of the total. Furthermore, every vertex in this component can be linked to any other vertex by a chain of no more than 20 calls.  $\blacktriangleleft$



Sometimes the removal of a vertex and all edges incident with it produces a subgraph with more connected components than in the original graph. Such vertices are called **cut vertices** (or **articulation points**). The removal of a cut vertex from a connected graph produces a subgraph that is not connected. Analogously, an edge whose removal produces a graph with more connected components than in the original graph is called a **cut edge** or **bridge**.

**EXAMPLE 8** Find the cut vertices and cut edges in the graph  $G$  shown in Figure 4.

*Solution:* The cut vertices of  $G$  are  $b$ ,  $c$ , and  $e$ . The removal of one of these vertices (and its adjacent edges) disconnects the graph. The cut edges are  $\{a, b\}$  and  $\{c, e\}$ . Removing either one of these edges disconnects  $G$ . ◀

## Connectedness in Directed Graphs

There are two notions of connectedness in directed graphs, depending on whether the directions of the edges are considered.

**DEFINITION 4** A directed graph is *strongly connected* if there is a path from  $a$  to  $b$  and from  $b$  to  $a$  whenever  $a$  and  $b$  are vertices in the graph.

For a directed graph to be strongly connected there must be a sequence of directed edges from any vertex in the graph to any other vertex. A directed graph can fail to be strongly connected but still be in “one piece.” Definition 5 makes this notion precise.

**DEFINITION 5** A directed graph is *weakly connected* if there is a path between every two vertices in the underlying undirected graph.

That is, a directed graph is weakly connected if and only if there is always a path between two vertices when the directions of the edges are disregarded. Clearly, any strongly connected directed graph is also weakly connected.

**EXAMPLE 9** Are the directed graphs  $G$  and  $H$  shown in Figure 5 strongly connected? Are they weakly connected?

*Solution:*  $G$  is strongly connected because there is a path between any two vertices in this directed graph (the reader should verify this). Hence,  $G$  is also weakly connected. The graph  $H$  is not strongly connected. There is no directed path from  $a$  to  $b$  in this graph. However,  $H$  is weakly connected, because there is a path between any two vertices in the underlying undirected graph of  $H$  (the reader should verify this). ◀

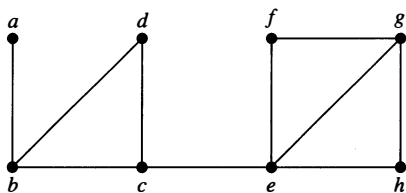


FIGURE 4 The Graph  $G$ .

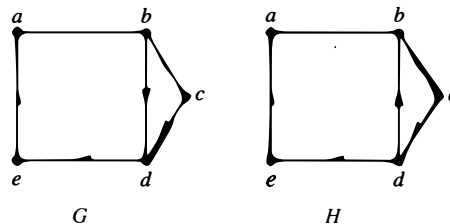


FIGURE 5 The Directed Graphs  $G$  and  $H$ .

The subgraphs of a directed graph  $G$  that are strongly connected but not contained in larger strongly connected subgraphs, that is, the maximal strongly connected subgraphs, are called the **strongly connected components** or **strong components** of  $G$ .

**EXAMPLE 10** The graph  $H$  in Figure 5 has three strongly connected components, consisting of the vertex  $a$ ; the vertex  $e$ ; and the graph consisting of the vertices  $b, c$ , and  $d$  and edges  $(b, c)$ ,  $(c, d)$ , and  $(d, b)$ . ◀

**EXAMPLE 11** **The Strongly Connected Components of the Web Graph** The Web graph introduced in Example 8 of Section 9.1 represents Web pages with vertices and links with directed edges. A snapshot of the Web in 1999 produced a Web graph with over 200 million vertices and over 1.5 billion edges. (See [Br00] for details.) The underlying undirected graph of this Web graph is not connected and has a connected component that includes approximately 90% of the vertices in the graph. The subgraph of the original directed graph corresponding to this connected component of the underlying undirected graph (that is, with the same vertices and all directed edges connecting vertices in this graph) has one very large strongly connected component and many small ones. The former is called the **giant strongly connected component (GSCC)** of the directed graph. A Web page in this component can be reached following links starting at any other page in this component. The GSCC in the Web graph produced by this study was found to have over 53 million vertices. The remaining vertices in the large connected component of the undirected graph represent three different types of Web pages: pages that can be reached from a page in the GSCC, but do not link back to these pages following a series of links; pages that link back to pages in the GSCC following a series of links, but cannot be reached by following links on pages in the GSCC; and pages that cannot reach pages in the GSCC and cannot be reached from pages in the GSCC following a series of links. In this study, each of these three other sets was found to have approximately 44 million vertices. (It is rather surprising that these three sets are close to the same size.) ◀



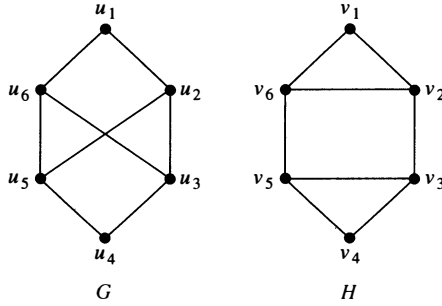
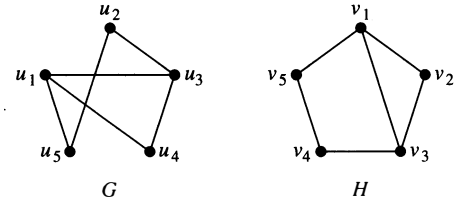
## Paths and Isomorphism

There are several ways that paths and circuits can help determine whether two graphs are isomorphic. For example, the existence of a simple circuit of a particular length is a useful invariant that can be used to show that two graphs are not isomorphic. In addition, paths can be used to construct mappings that may be isomorphisms.

As we mentioned, a useful isomorphic invariant for simple graphs is the existence of a simple circuit of length  $k$ , where  $k$  is a positive integer greater than 2. (The proof that this is an invariant is left as Exercise 50 at the end of this section.) Example 12 illustrates how this invariant can be used to show that two graphs are not isomorphic.

**EXAMPLE 12** Determine whether the graphs  $G$  and  $H$  shown in Figure 6 are isomorphic.

*Solution:* Both  $G$  and  $H$  have six vertices and eight edges. Each has four vertices of degree three, and two vertices of degree two. So, the three invariants—number of vertices, number of edges, and degrees of vertices—all agree for the two graphs. However,  $H$  has a simple circuit of length three, namely,  $v_1, v_2, v_6, v_1$ , whereas  $G$  has no simple circuit of length three, as can be determined by inspection (all simple circuits in  $G$  have length at least four). Because the existence of a simple circuit of length three is an isomorphic invariant,  $G$  and  $H$  are not isomorphic. ◀

FIGURE 6 The Graphs  $G$  and  $H$ .FIGURE 7 The Graphs  $G$  and  $H$ .

We have shown how the existence of a type of path, namely, a simple circuit of a particular length, can be used to show that two graphs are not isomorphic. We can also use paths to find mappings that are potential isomorphisms.

**EXAMPLE 13** Determine whether the graphs  $G$  and  $H$  shown in Figure 7 are isomorphic.

*Solution:* Both  $G$  and  $H$  have five vertices and six edges, both have two vertices of degree three and three vertices of degree two, and both have a simple circuit of length three, a simple circuit of length four, and a simple circuit of length five. Because all these isomorphic invariants agree,  $G$  and  $H$  may be isomorphic. To find a possible isomorphism, we can follow paths that go through all vertices so that the corresponding vertices in the two graphs have the same degree. For example, the paths  $u_1, u_4, u_3, u_2, u_5$  in  $G$  and  $v_3, v_2, v_1, v_5, v_4$  in  $H$  both go through every vertex in the graph; start at a vertex of degree three; go through vertices of degrees two, three, and two, respectively; and end at a vertex of degree two. By following these paths through the graphs, we define the mapping  $f$  with  $f(u_1) = v_3$ ,  $f(u_4) = v_2$ ,  $f(u_3) = v_1$ ,  $f(u_2) = v_5$ , and  $f(u_5) = v_4$ . The reader can show that  $f$  is an isomorphism, so  $G$  and  $H$  are isomorphic, either by showing that  $f$  preserves edges or by showing that with the appropriate orderings of vertices the adjacency matrices of  $G$  and  $H$  are the same. ◀

## Counting Paths Between Vertices

The number of paths between two vertices in a graph can be determined using its adjacency matrix.

**THEOREM 2** Let  $G$  be a graph with adjacency matrix  $A$  with respect to the ordering  $v_1, v_2, \dots, v_n$  (with directed or undirected edges, with multiple edges and loops allowed). The number of different paths of length  $r$  from  $v_i$  to  $v_j$ , where  $r$  is a positive integer, equals the  $(i, j)$ th entry of  $A^r$ .

*Proof:* The theorem will be proved using mathematical induction. Let  $G$  be a graph with adjacency matrix  $A$  (assuming an ordering  $v_1, v_2, \dots, v_n$  of the vertices of  $G$ ). The number of paths from  $v_i$  to  $v_j$  of length 1 is the  $(i, j)$ th entry of  $A$ , because this entry is the number of edges from  $v_i$  to  $v_j$ .

Assume that the  $(i, j)$ th entry of  $A^r$  is the number of different paths of length  $r$  from  $v_i$  to  $v_j$ . This is the induction hypothesis. Because  $A^{r+1} = A^r A$ , the  $(i, j)$ th entry of  $A^{r+1}$  equals

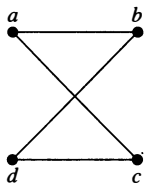
$$b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj},$$



where  $b_{ik}$  is the  $(i, k)$ th entry of  $A^r$ . By the induction hypothesis,  $b_{ik}$  is the number of paths of length  $r$  from  $v_i$  to  $v_k$ .

A path of length  $r + 1$  from  $v_i$  to  $v_j$  is made up of a path of length  $r$  from  $v_i$  to some intermediate vertex  $v_k$ , and an edge from  $v_k$  to  $v_j$ . By the product rule for counting, the number of such paths is the product of the number of paths of length  $r$  from  $v_i$  to  $v_k$ , namely,  $b_{ik}$ , and the number of edges from  $v_k$  to  $v_j$ , namely,  $a_{kj}$ . When these products are added for all possible intermediate vertices  $v_k$ , the desired result follows by the sum rule for counting. ◀

**EXAMPLE 14** How many paths of length four are there from  $a$  to  $d$  in the simple graph  $G$  in Figure 8?



*Solution:* The adjacency matrix of  $G$  (ordering the vertices as  $a, b, c, d$ ) is

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

**FIGURE 8** The Graph  $G$ .

Hence, the number of paths of length four from  $a$  to  $d$  is the  $(1, 4)$ th entry of  $A^4$ . Because

$$A^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix},$$



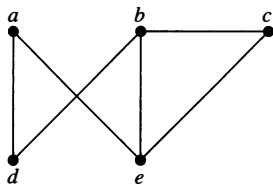
there are exactly eight paths of length four from  $a$  to  $d$ . By inspection of the graph, we see that  $a, b, a, b, d$ ;  $a, b, a, c, d$ ;  $a, b, d, b, d$ ;  $a, b, d, c, d$ ;  $a, c, a, b, d$ ;  $a, c, a, c, d$ ;  $a, c, d, b, d$ ; and  $a, c, d, c, d$  are the eight paths from  $a$  to  $d$ . ◀

Theorem 2 can be used to find the length of the shortest path between two vertices of a graph (see Exercise 46), and it can also be used to determine whether a graph is connected (see Exercises 51 and 52).

## Exercises

1. Does each of these lists of vertices form a path in the following graph? Which paths are simple? Which are circuits? What are the lengths of those that are paths?

- a)  $a, e, b, c, b$                       b)  $a, e, a, d, b, c, a$   
c)  $e, b, a, d, b, e$                       d)  $c, b, d, a, e, c$

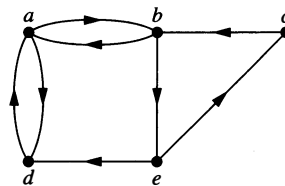


2. Does each of these lists of vertices form a path in the following graph? Which paths are simple? Which are circuits? What are the lengths of those that are paths?

- a)  $a, b, e, c, b$                       b)  $a, d, a, d, a$

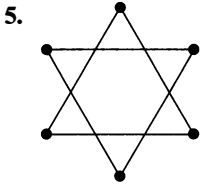
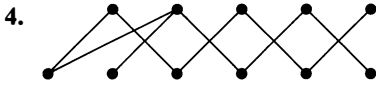
c)  $a, d, b, e, a$

d)  $a, b, e, c, b, d, a$

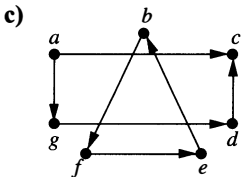
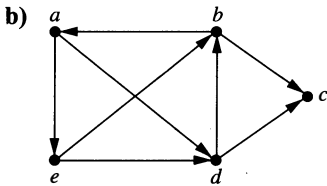
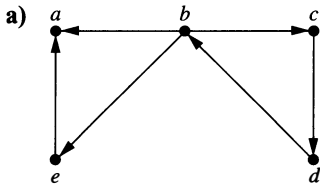


In Exercises 3–5 determine whether the given graph is connected.

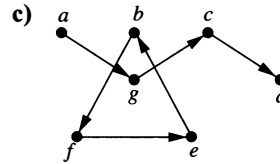
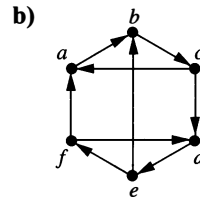
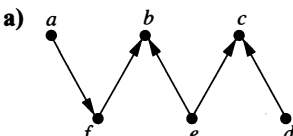




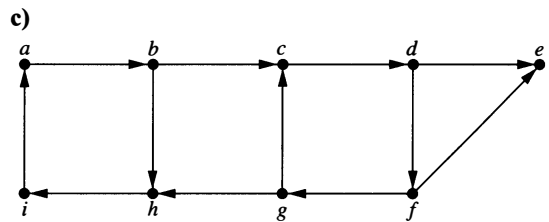
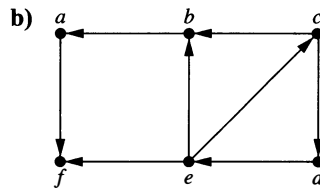
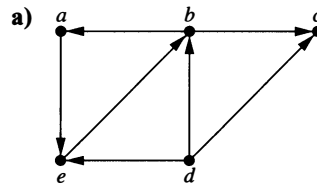
6. How many connected components does each of the graphs in Exercises 3–5 have? For each graph find each of its connected components.
7. What do the connected components of acquaintanceship graphs represent?
8. What do the connected components of a collaboration graph represent?
9. Explain why in the collaboration graph of mathematicians a vertex representing a mathematician is in the same connected component as the vertex representing Paul Erdős if and only if that mathematician has a finite Erdős number.
10. In the Hollywood graph (see Example 4 in Section 9.1), when is the vertex representing an actor in the same connected component as the vertex representing Kevin Bacon?
11. Determine whether each of these graphs is strongly connected and if not, whether it is weakly connected.



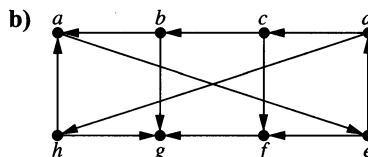
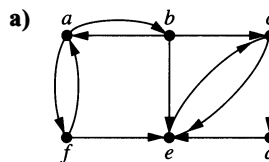
12. Determine whether each of these graphs is strongly connected and if not, whether it is weakly connected.

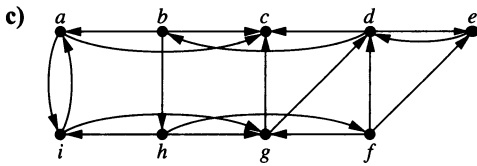


13. What do the strongly connected components of a telephone call graph represent?
14. Find the strongly connected components of each of these graphs.

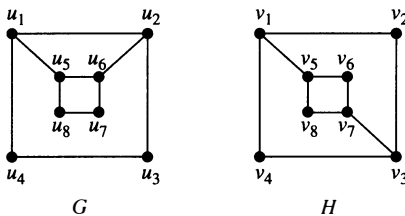


15. Find the strongly connected components of each of these graphs.

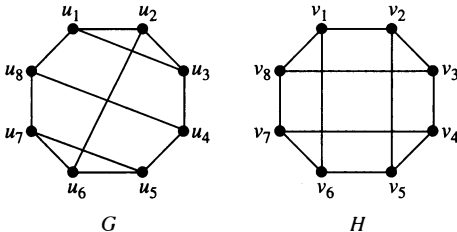




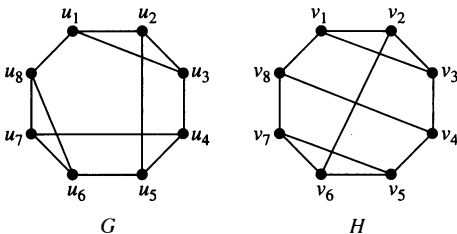
16. Show that all vertices visited in a directed path connecting two vertices in the same strongly connected component of a directed graph are also in this strongly connected component.
17. Find the number of paths of length  $n$  between two different vertices in  $K_4$  if  $n$  is  
 a) 2.    b) 3.    c) 4.    d) 5.
18. Use paths either to show that these graphs are not isomorphic or to find an isomorphism between these graphs.



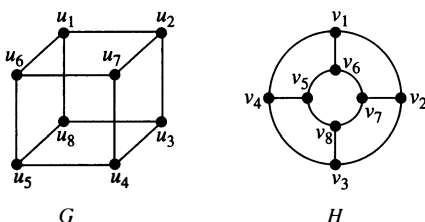
19. Use paths either to show that these graphs are not isomorphic or to find an isomorphism between them.



20. Use paths either to show that these graphs are not isomorphic or to find an isomorphism between them.

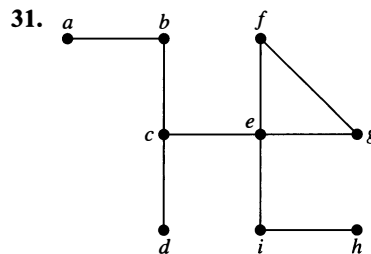
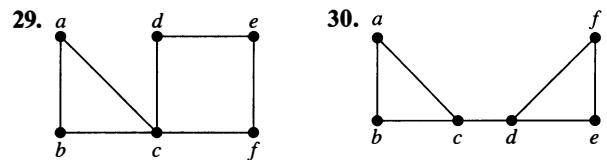


21. Use paths either to show that these graphs are not isomorphic or to find an isomorphism between them.



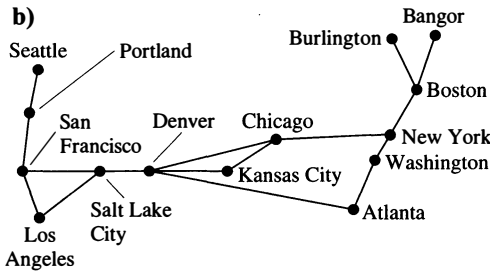
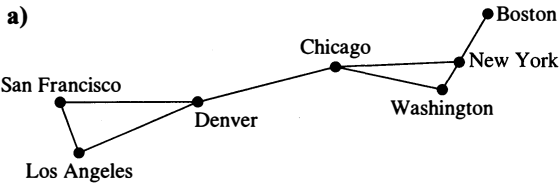
22. Find the number of paths of length  $n$  between any two adjacent vertices in  $K_{3,3}$  for the values of  $n$  in Exercise 17.
23. Find the number of paths of length  $n$  between any two non-adjacent vertices in  $K_{3,3}$  for the values of  $n$  in Exercise 17.
24. Find the number of paths between  $c$  and  $d$  in the graph in Figure 1 of length  
 a) 2.    b) 3.    c) 4.    d) 5.    e) 6.    f) 7.
25. Find the number of paths from  $a$  to  $e$  in the directed graph in Exercise 2 of length  
 a) 2.    b) 3.    c) 4.    d) 5.    e) 6.    f) 7.
- \*26. Show that every connected graph with  $n$  vertices has at least  $n - 1$  edges.
27. Let  $G = (V, E)$  be a simple graph. Let  $R$  be the relation on  $V$  consisting of pairs of vertices  $(u, v)$  such that there is a path from  $u$  to  $v$  or such that  $u = v$ . Show that  $R$  is an equivalence relation.
- \*28. Show that in every simple graph there is a path from any vertex of odd degree to some other vertex of odd degree.

In Exercises 29–31 find all the cut vertices of the given graph.



32. Find all the cut edges in the graphs in Exercises 29–31.
- \*33. Suppose that  $v$  is an endpoint of a cut edge. Prove that  $v$  is a cut vertex if and only if this vertex is not pendant.
- \*34. Show that a vertex  $c$  in the connected simple graph  $G$  is a cut vertex if and only if there are vertices  $u$  and  $v$ , both different from  $c$ , such that every path between  $u$  and  $v$  passes through  $c$ .
- \*35. Show that a simple graph with at least two vertices has at least two vertices that are not cut vertices.
- \*36. Show that an edge in a simple graph is a cut edge if and only if this edge is not part of any simple circuit in the graph.
37. A communications link in a network should be provided with a backup link if its failure makes it impossible for some message to be sent. For each of the communications

networks shown here in (a) and (b), determine those links that should be backed up.



A **vertex basis** in a directed graph is a set of vertices such that there is a path to every vertex in the directed graph not in the set from some vertex in this set and there is no path from any vertex in the set to another vertex in the set.

38. Find a vertex basis for each of the directed graphs in Exercises 7–9 of Section 9.2.
39. What is the significance of a vertex basis in an influence graph (described in Example 3 of Section 9.1)? Find a vertex basis in the influence graph in this example.
40. Show that if a connected simple graph  $G$  is the union of the graphs  $G_1$  and  $G_2$ , then  $G_1$  and  $G_2$  have at least one common vertex.
- \*41. Show that if a simple graph  $G$  has  $k$  connected components and these components have  $n_1, n_2, \dots, n_k$  vertices, respectively, then the number of edges of  $G$  does not exceed

$$\sum_{i=1}^k C(n_i, 2).$$

- \*42. Use Exercise 41 to show that a simple graph with  $n$  vertices and  $k$  connected components has at most  $(n - k)(n - k + 1)/2$  edges. [Hint: First show that

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

where  $n_i$  is the number of vertices in the  $i$ th connected component.]

- \*43. Show that a simple graph  $G$  with  $n$  vertices is connected if it has more than  $(n - 1)(n - 2)/2$  edges.
44. Describe the adjacency matrix of a graph with  $n$  connected components when the vertices of the graph are listed so that vertices in each connected component are listed successively.

45. How many nonisomorphic connected simple graphs are there with  $n$  vertices when  $n$  is

a) 2?      b) 3?      c) 4?      d) 5?

46. Explain how Theorem 2 can be used to find the length of the shortest path from a vertex  $v$  to a vertex  $w$  in a graph.
47. Use Theorem 2 to find the length of the shortest path between  $a$  and  $f$  in the graph in Figure 1.
48. Use Theorem 2 to find the length of the shortest path from  $a$  to  $c$  in the directed graph in Exercise 2.
49. Let  $P_1$  and  $P_2$  be two simple paths between the vertices  $u$  and  $v$  in the simple graph  $G$  that do not contain the same set of edges. Show that there is a simple circuit in  $G$ .
50. Show that the existence of a simple circuit of length  $k$ , where  $k$  is a positive integer greater than 2, is an isomorphic invariant.
51. Explain how Theorem 2 can be used to determine whether a graph is connected.
52. Use Exercise 51 to show that the graph  $G_1$  in Figure 2 is connected whereas the graph  $G_2$  in that figure is not connected.
53. Show that a simple graph  $G$  is bipartite if and only if it has no circuits with an odd number of edges.
54. In an old puzzle attributed to Alcuin of York (735–804), a farmer needs to carry a wolf, a goat, and a cabbage across a river. The farmer only has a small boat, which can carry the farmer and only one object (an animal or a vegetable). He can cross the river repeatedly. However, if the farmer is on the other shore, the wolf will eat the goat, and, similarly, the goat will eat the cabbage. We can describe each state by listing what is on each shore. For example, we can use the pair  $(FG, WC)$  for the state where the farmer and goat are on the first shore and the wolf and cabbage are on the other shore. [The symbol  $\emptyset$  is used when nothing is on a shore, so that  $(FWGC, \emptyset)$  is the initial state.]
- a) Find all allowable states of the puzzle, where neither the wolf and the goat nor the goat and the cabbage are left on the same shore without the farmer.
- b) Construct a graph such that each vertex of this graph represents an allowable state and the vertices representing two allowable states are connected by an edge if it is possible to move from one state to the other using one trip of the boat.
- c) Explain why finding a path from the vertex representing  $(FWGC, \emptyset)$  to the vertex representing  $(\emptyset, FWGC)$  solves the puzzle.
- d) Find two different solutions of the puzzle, each using seven crossings.
- e) Suppose that the farmer must pay a toll of one dollar whenever he crosses the river with an animal. Which solution of the puzzle should the farmer use to pay the least total toll?
- \*55. Use a graph model and a path in your graph, as in Exercise 54, to solve the **jealous husbands problem**. Two married couples, each a husband and a wife, want to cross

a river. They can only use a boat that can carry one or two people from one shore to the other shore. Each husband is extremely jealous and is not willing to leave his wife with the other husband, either in the boat or on shore. How can these four people reach the opposite shore?

56. Suppose that you have a three-gallon jug and a five-gallon jug, and you may fill either jug from a water tap, you may

empty either jug, and you may transfer water from either jug into the other jug. Use a path in a directed graph model to show that you can end up with a jug containing exactly one gallon. [Hint: Use an ordered pair  $(a, b)$  to indicate how much water is in each of the jugs and represent these ordered pairs by vertices. Add edges corresponding to the allowable operations with the jugs.]

## 9.5 Euler and Hamilton Paths

### Introduction

Can we travel along the edges of a graph starting at a vertex and returning to it by traversing each edge of the graph exactly once? Similarly, can we travel along the edges of a graph starting at a vertex and returning to it while visiting each vertex of the graph exactly once? Although these questions seem to be similar, the first question, which asks whether a graph has an *Euler circuit*, can be easily answered simply by examining the degrees of the vertices of the graph, while the second question, which asks whether a graph has a *Hamilton circuit*, is quite difficult to solve for most graphs. In this section we will study these questions and discuss the difficulty of solving them. Although both questions have many practical applications in many different areas, both arose in old puzzles. We will learn about these old puzzles as well as modern practical applications.

### Euler Paths and Circuits



The town of Königsberg, Prussia (now called Kaliningrad and part of the Russian republic), was divided into four sections by the branches of the Pregel River. These four sections included the two regions on the banks of the Pregel, Kneiphof Island, and the region between the two branches of the Pregel. In the eighteenth century seven bridges connected these regions. Figure 1 depicts these regions and bridges.

The townspeople took long walks through town on Sundays. They wondered whether it was possible to start at some location in the town, travel across all the bridges without crossing any bridge twice, and return to the starting point.

The Swiss mathematician Leonhard Euler solved this problem. His solution, published in 1736, may be the first use of graph theory. Euler studied this problem using the multigraph obtained when the four regions are represented by vertices and the bridges by edges. This multigraph is shown in Figure 2.

The problem of traveling across every bridge without crossing any bridge more than once can be rephrased in terms of this model. The question becomes: Is there a simple circuit in this multigraph that contains every edge?

**DEFINITION 1** An *Euler circuit* in a graph  $G$  is a simple circuit containing every edge of  $G$ . An *Euler path* in  $G$  is a simple path containing every edge of  $G$ .

Examples 1 and 2 illustrate the concept of Euler circuits and paths.

**EXAMPLE 1** Which of the undirected graphs in Figure 3 have an Euler circuit? Of those that do not, which have an Euler path?

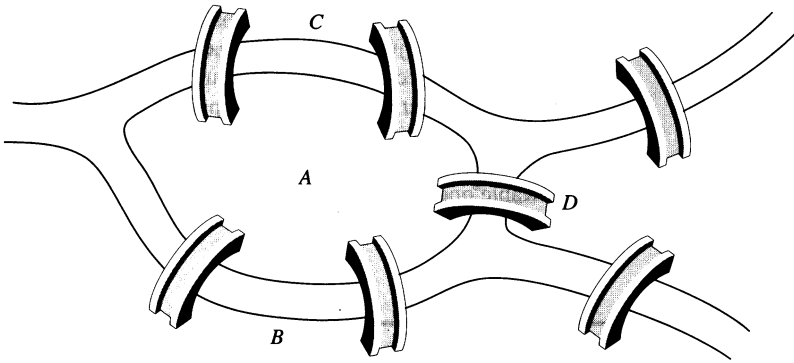


FIGURE 1 The Seven Bridges of Königsberg.

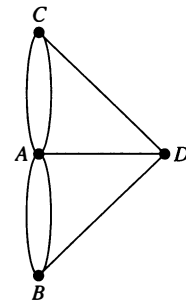


FIGURE 2 Multigraph Model of the Town of Königsberg.

*Solution:* The graph  $G_1$  has an Euler circuit, for example,  $a, e, c, d, e, b, a$ . Neither of the graphs  $G_2$  or  $G_3$  has an Euler circuit (the reader should verify this). However,  $G_3$  has an Euler path, namely,  $a, c, d, e, b, d, a, b$ .  $G_2$  does not have an Euler path (as the reader should verify). ◀

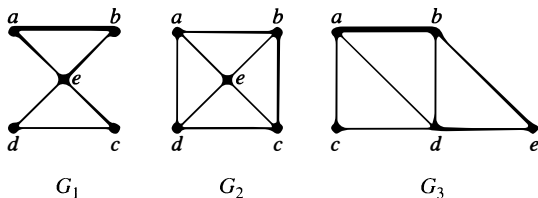
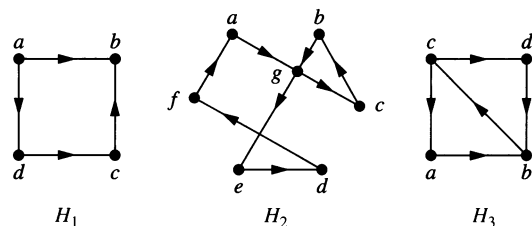
**EXAMPLE 2** Which of the directed graphs in Figure 4 have an Euler circuit? Of those that do not, which have an Euler path?



*Solution:* The graph  $H_2$  has an Euler circuit, for example,  $a, g, c, b, g, e, d, f, a$ . Neither  $H_1$  nor  $H_3$  has an Euler circuit (as the reader should verify).  $H_3$  has an Euler path, namely,  $c, a, b, c, d, b$ , but  $H_1$  does not (as the reader should verify). ◀

**NECESSARY AND SUFFICIENT CONDITIONS FOR EULER CIRCUITS AND PATHS**  
There are simple criteria for determining whether a multigraph has an Euler circuit or an Euler path. Euler discovered them when he solved the famous Königsberg bridge problem. We will assume that all graphs discussed in this section have a finite number of vertices and edges.

What can we say if a connected multigraph has an Euler circuit? What we can show is that every vertex must have even degree. To do this, first note that an Euler circuit begins with a vertex  $a$  and continues with an edge incident with  $a$ , say  $\{a, b\}$ . The edge  $\{a, b\}$  contributes one to  $\deg(a)$ . Each time the circuit passes through a vertex it contributes two to the vertex's degree, because the circuit enters via an edge incident with this vertex and leaves via another such edge. Finally, the circuit terminates where it started, contributing one to  $\deg(a)$ . Therefore,  $\deg(a)$  must be even, because the circuit contributes one when it begins, one when it ends, and two every time it passes through  $a$  (if it ever does). A vertex other than  $a$  has even degree because the circuit contributes two to its degree each time it passes through the vertex. We conclude that if a connected graph has an Euler circuit, then every vertex must have even degree.

FIGURE 3 The Undirected Graphs  $G_1$ ,  $G_2$ , and  $G_3$ .FIGURE 4 The Directed Graphs  $H_1$ ,  $H_2$ , and  $H_3$ .

Is this necessary condition for the existence of an Euler circuit also sufficient? That is, must an Euler circuit exist in a connected multigraph if all vertices have even degree? This question can be settled affirmatively with a construction.

Suppose that  $G$  is a connected multigraph with at least two vertices and the degree of every vertex of  $G$  is even. We will form a simple circuit that begins at an arbitrary vertex  $a$  of  $G$ . Let  $x_0 = a$ . First, we arbitrarily choose an edge  $\{x_0, x_1\}$  incident with  $a$  which is possible because  $G$  is connected. We continue by building a simple path  $\{x_0, x_1\}, \{x_1, x_2\}, \dots, \{x_{n-1}, x_n\}$ , adding edges to the path until we cannot add another edge to the path. This happens when we reach a vertex for which we have already included all edges incident with that vertex in the path. For instance, in the graph  $G$  in Figure 5 we begin at  $a$  and choose in succession the edges  $\{a, f\}$ ,  $\{f, c\}$ ,  $\{c, b\}$ , and  $\{b, a\}$ .

The path we have constructed must terminate because the graph has a finite number of edges, so we are guaranteed to eventually reach a vertex for which no edges are available to add to the path. The path begins at  $a$  with an edge of the form  $\{a, x\}$ , and we now show that it must terminate at  $a$  with an edge of the form  $\{y, a\}$ . To see that the path must terminate at  $a$ , note that each time the path goes through a vertex with even degree, it uses only one edge to enter this vertex, so because the degree must be at least two, at least one edge remains for the path to leave the vertex. Furthermore, every time we enter and leave a vertex of even degree, there are an even number of edges incident with this vertex that we have not yet used in our path. Consequently, as we form the path, every time we enter a vertex other than  $a$ , we can leave it. This means that path can end only at  $a$ . Next, note that the path we have constructed may use all the edges of the graph, or it may not if we have returned to  $a$  for the last time before using all the edges.

An Euler circuit has been constructed if all the edges have been used. Otherwise, consider the subgraph  $H$  obtained from  $G$  by deleting the edges already used and vertices that are not incident with any remaining edges. When we delete the circuit  $a, f, c, b, a$  from the graph in Figure 5, we obtain the subgraph labeled as  $H$ .

Because  $G$  is connected,  $H$  has at least one vertex in common with the circuit that has been deleted. Let  $w$  be such a vertex. (In our example,  $c$  is the vertex.)

Every vertex in  $H$  has even degree (because in  $G$  all vertices had even degree, and for each vertex, pairs of edges incident with this vertex have been deleted to form  $H$ ). Note that  $H$  may not be connected. Beginning at  $w$ , construct a simple path in  $H$  by choosing edges as long as possible, as was done in  $G$ . This path must terminate at  $w$ . For instance, in Figure 5,  $c, d, e, c$  is a path in  $H$ . Next, form a circuit in  $G$  by splicing the circuit in  $H$  with the original circuit in  $G$  (this can be done because  $w$  is one of the vertices in this circuit). When this is done in the graph in Figure 5, we obtain the circuit  $a, f, c, d, e, c, b, a$ .

Continue this process until all edges have been used. (The process must terminate because there are only a finite number of edges in the graph.) This produces an Euler circuit. The



**LEONHARD EULER (1707–1783)** Leonhard Euler was the son of a Calvinist minister from the vicinity of Basel, Switzerland. At 13 he entered the University of Basel, pursuing a career in theology, as his father wished. At the university Euler was tutored by Johann Bernoulli of the famous Bernoulli family of mathematicians. His interest and skills led him to abandon his theological studies and take up mathematics. Euler obtained his master's degree in philosophy at the age of 16. In 1727 Peter the Great invited him to join the Academy at St. Petersburg. In 1741 he moved to the Berlin Academy, where he stayed until 1766. He then returned to St. Petersburg, where he remained for the rest of his life.

Euler was incredibly prolific, contributing to many areas of mathematics, including number theory, combinatorics, and analysis, as well as its applications to such areas as music and naval architecture. He wrote over 1100 books and papers and left so much unpublished work that it took 47 years after he died for all his work to be published. During his life his papers accumulated so quickly that he kept a large pile of articles awaiting publication. The Berlin Academy published the papers on top of this pile so later results were often published before results they depended on or superseded. Euler had 13 children and was able to continue his work while a child or two bounced on his knees. He was blind for the last 17 years of his life, but because of his fantastic memory this did not diminish his mathematical output. The project of publishing his collected works, undertaken by the Swiss Society of Natural Science, is ongoing and will require more than 75 volumes.

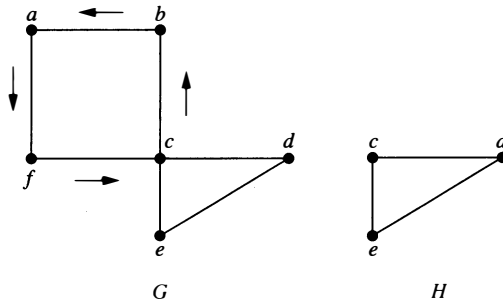


FIGURE 5 Constructing an Euler Circuit in  $G$ .

construction shows that if the vertices of a connected multigraph all have even degree, then the graph has an Euler circuit.

We summarize these results in Theorem 1.

**THEOREM 1** A connected multigraph with at least two vertices has an Euler circuit if and only if each of its vertices has even degree.

We can now solve the Königsberg bridge problem. Because the multigraph representing these bridges, shown in Figure 2, has four vertices of odd degree, it does not have an Euler circuit. There is no way to start at a given point, cross each bridge exactly once, and return to the starting point.

Algorithm 1 gives the constructive procedure for finding Euler circuits given in the discussion preceding Theorem 1. (Because the circuits in the procedure are chosen arbitrarily, there is some ambiguity. We will not bother to remove this ambiguity by specifying the steps of the procedure more precisely.)

**ALGORITHM 1** Constructing Euler Circuits.

```

procedure Euler( $G$ : connected multigraph with all vertices of
    even degree)
  circuit := a circuit in  $G$  beginning at an arbitrarily chosen
    vertex with edges successively added to form a path that
    returns to this vertex
   $H$  :=  $G$  with the edges of this circuit removed
  while  $H$  has edges
  begin
    subcircuit := a circuit in  $H$  beginning at a vertex in  $H$  that
      also is an endpoint of an edge of circuit
     $H$  :=  $H$  with edges of subcircuit and all isolated vertices
      removed
    circuit := circuit with subcircuit inserted at the appropriate
      vertex
  end {circuit is an Euler circuit}
  
```

Example 3 shows how Euler paths and circuits can be used to solve a type of puzzle.



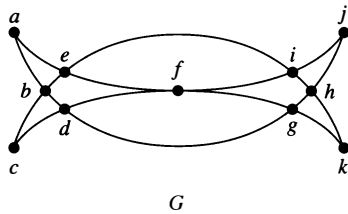


FIGURE 6 Mohammed's Scimitars.

**EXAMPLE 3** Many puzzles ask you to draw a picture in a continuous motion without lifting a pencil so that no part of the picture is retraced. We can solve such puzzles using Euler circuits and paths. For example, can *Mohammed's scimitars*, shown in Figure 6, be drawn in this way, where the drawing begins and ends at the same point?

*Solution:* We can solve this problem because the graph  $G$  shown in Figure 6 has an Euler circuit. It has such a circuit because all its vertices have even degree. We will use Algorithm 1 to construct an Euler circuit. First, we form the circuit  $a, b, d, c, b, e, i, f, e, a$ . We obtain the subgraph  $H$  by deleting the edges in this circuit and all vertices that become isolated when these edges are removed. Then we form the circuit  $d, g, h, j, i, h, k, g, f, d$  in  $H$ . After forming this circuit we have used all edges in  $G$ . Splicing this new circuit into the first circuit at the appropriate place produces the Euler circuit  $a, b, d, g, h, j, i, h, k, g, f, d, c, b, e, i, f, e, a$ . This circuit gives a way to draw the scimitars without lifting the pencil or retracing part of the picture. ◀

Another algorithm for constructing Euler circuits, called Fleury's algorithm, is described in the exercises at the end of this section.

We will now show that a connected multigraph has an Euler path (and not an Euler circuit) if and only if it has exactly two vertices of odd degree. First, suppose that a connected multigraph does have an Euler path from  $a$  to  $b$ , but not an Euler circuit. The first edge of the path contributes one to the degree of  $a$ . A contribution of two to the degree of  $a$  is made every time the path passes through  $a$ . The last edge in the path contributes one to the degree of  $b$ . Every time the path goes through  $b$  there is a contribution of two to its degree. Consequently, both  $a$  and  $b$  have odd degree. Every other vertex has even degree, because the path contributes two to the degree of a vertex whenever it passes through it.

Now consider the converse. Suppose that a graph has exactly two vertices of odd degree, say  $a$  and  $b$ . Consider the larger graph made up of the original graph with the addition of an edge  $\{a, b\}$ . Every vertex of this larger graph has even degree, so there is an Euler circuit. The removal of the new edge produces an Euler path in the original graph. Theorem 2 summarizes these results.

**THEOREM 2** A connected multigraph has an Euler path but not an Euler circuit if and only if it has exactly two vertices of odd degree.

**EXAMPLE 4** Which graphs shown in Figure 7 have an Euler path?

*Solution:*  $G_1$  contains exactly two vertices of odd degree, namely,  $b$  and  $d$ . Hence, it has an Euler path that must have  $b$  and  $d$  as its endpoints. One such Euler path is  $d, a, b, c, d, b$ . Similarly,

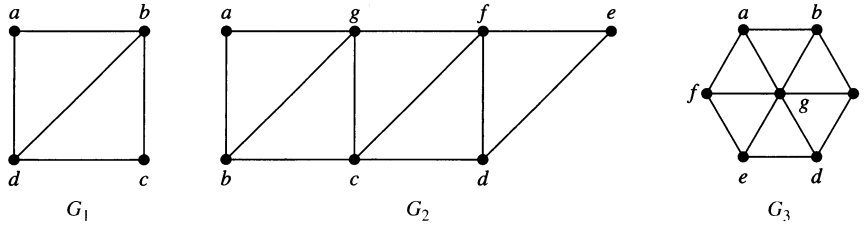


FIGURE 7 Three Undirected Graphs.

$G_2$  has exactly two vertices of odd degree, namely,  $b$  and  $d$ . So it has an Euler path that must have  $b$  and  $d$  as endpoints. One such Euler path is  $b, a, g, f, e, d, c, g, b, c, f, d$ .  $G_3$  has no Euler path because it has six vertices of odd degree. ◀

Returning to eighteenth-century Königsberg, is it possible to start at some point in the town, travel across all the bridges, and end up at some other point in town? This question can be answered by determining whether there is an Euler path in the multigraph representing the bridges in Königsberg. Because there are four vertices of odd degree in this multigraph, there is no Euler path, so such a trip is impossible.

Necessary and sufficient conditions for Euler paths and circuits in directed graphs are discussed in the exercises at the end of this section.



Euler paths and circuits can be used to solve many practical problems. For example, many applications ask for a path or circuit that traverses each street in a neighborhood, each road in a transportation network, each connection in a utility grid, or each link in a communications network exactly once. Finding an Euler path or circuit in the appropriate graph model can solve such problems. For example, if a postman can find an Euler path in the graph that represents the streets the postman needs to cover, this path produces a route that traverses each street of the route exactly once. If no Euler path exists, some streets will have to be traversed more than once. This problem is known as the *Chinese postman problem* in honor of Guan Meigu, who posed it in 1962. See [MiRo91] for more information on the solution of the Chinese postman problem when no Euler path exists.

Among the other areas where Euler circuits and paths are applied is in layout of circuits, in network multicasting, and in molecular biology, where Euler paths are used in the sequencing of DNA.

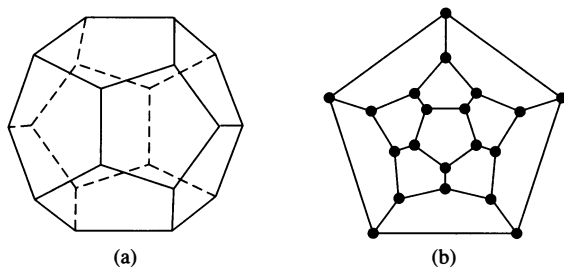
## Hamilton Paths and Circuits



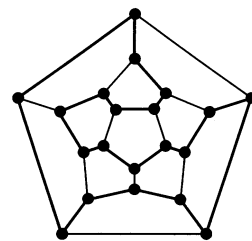
We have developed necessary and sufficient conditions for the existence of paths and circuits that contain every edge of a multigraph exactly once. Can we do the same for simple paths and circuits that contain every vertex of the graph exactly once?

### DEFINITION 2

A simple path in a graph  $G$  that passes through every vertex exactly once is called a *Hamilton path*, and a simple circuit in a graph  $G$  that passes through every vertex exactly once is called a *Hamilton circuit*. That is, the simple path  $x_0, x_1, \dots, x_{n-1}, x_n$  in the graph  $G = (V, E)$  is a Hamilton path if  $V = \{x_0, x_1, \dots, x_{n-1}, x_n\}$  and  $x_i \neq x_j$  for  $0 \leq i < j \leq n$ , and the simple circuit  $x_0, x_1, \dots, x_{n-1}, x_n, x_0$  (with  $n > 0$ ) is a Hamilton circuit if  $x_0, x_1, \dots, x_{n-1}, x_n$  is a Hamilton path.



**FIGURE 8** Hamilton's "A Voyage Round the World" Puzzle.



**FIGURE 9** A Solution to the "A Voyage Round the World" Puzzle.

This terminology comes from a game, called the *Icosian puzzle*, invented in 1857 by the Irish mathematician Sir William Rowan Hamilton. It consisted of a wooden dodecahedron [a polyhedron with 12 regular pentagons as faces, as shown in Figure 8(a)], with a peg at each vertex of the dodecahedron, and string. The 20 vertices of the dodecahedron were labeled with different cities in the world. The object of the puzzle was to start at a city and travel along the edges of the dodecahedron, visiting each of the other 19 cities exactly once, and end back at the first city. The circuit traveled was marked off using the strings and pegs.

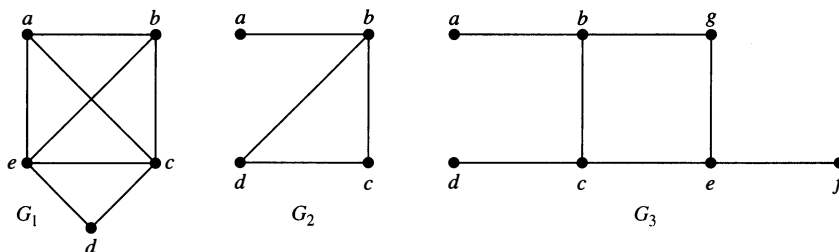
Because the author cannot supply each reader with a wooden solid with pegs and string, we will consider the equivalent question: Is there a circuit in the graph shown in Figure 8(b) that passes through each vertex exactly once? This solves the puzzle because this graph is isomorphic to the graph consisting of the vertices and edges of the dodecahedron. A solution of Hamilton's puzzle is shown in Figure 9.

**EXAMPLE 5** Which of the simple graphs in Figure 10 have a Hamilton circuit or, if not, a Hamilton path?

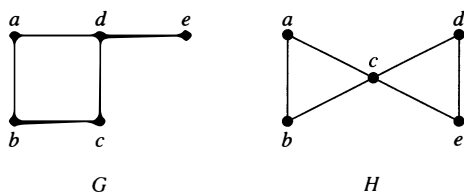


**Solution:**  $G_1$  has a Hamilton circuit:  $a, b, c, d, e, a$ . There is no Hamilton circuit in  $G_2$  (this can be seen by noting that any circuit containing every vertex must contain the edge  $\{a, b\}$  twice), but  $G_2$  does have a Hamilton path, namely,  $a, b, c, d$ .  $G_3$  has neither a Hamilton circuit nor a Hamilton path, because any path containing all vertices must contain one of the edges  $\{a, b\}$ ,  $\{e, f\}$ , and  $\{c, d\}$  more than once. ◀

Is there a simple way to determine whether a graph has a Hamilton circuit or path? At first, it might seem that there should be an easy way to determine this, because there is a simple way to answer the similar question of whether a graph has an Euler circuit. Surprisingly, there are no known simple necessary and sufficient criteria for the existence of Hamilton circuits. However, many theorems are known that give sufficient conditions for the existence of Hamilton circuits.



**FIGURE 10** Three Simple Graphs.

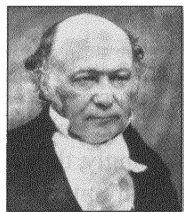


**FIGURE 11 Two Graphs That Do Not Have a Hamilton Circuit.**

Also, certain properties can be used to show that a graph has no Hamilton circuit. For instance, a graph with a vertex of degree one cannot have a Hamilton circuit, because in a Hamilton circuit, each vertex is incident with two edges in the circuit. Moreover, if a vertex in the graph has degree two, then both edges that are incident with this vertex must be part of any Hamilton circuit. Also, note that when a Hamilton circuit is being constructed and this circuit has passed through a vertex, then all remaining edges incident with this vertex, other than the two used in the circuit, can be removed from consideration. Furthermore, a Hamilton circuit cannot contain a smaller circuit within it.

**EXAMPLE 6** Show that neither graph displayed in Figure 11 has a Hamilton circuit.

*Solution:* There is no Hamilton circuit in  $G$  because  $G$  has a vertex of degree one, namely,  $e$ . Now consider  $H$ . Because the degrees of the vertices  $a$ ,  $b$ ,  $d$ , and  $e$  are all two, every edge incident with these vertices must be part of any Hamilton circuit. It is now easy to see that no Hamilton circuit can exist in  $H$ , for any Hamilton circuit would have to contain four edges incident with  $c$ , which is impossible. ◀



**WILLIAM ROWAN HAMILTON (1805–1865)** William Rowan Hamilton, the most famous Irish scientist ever to have lived, was born in 1805 in Dublin. His father was a successful lawyer, his mother came from a family noted for their intelligence, and he was a child prodigy. By the age of 3 he was an excellent reader and had mastered advanced arithmetic. Because of his brilliance, he was sent off to live with his uncle James, a noted linguist. By age 8 Hamilton had learned Latin, Greek, and Hebrew; by 10 he had also learned Italian and French and he began his study of oriental languages, including Arabic, Sanskrit, and Persian. During this period he took pride in knowing as many languages as his age. At 17, no longer devoted to learning new languages and having mastered calculus and much mathematical astronomy, he began original work in optics, and he also found an important mistake in Laplace's work on celestial mechanics.

Before entering Trinity College, Dublin, at 18, Hamilton had not attended school; rather, he received private tutoring. At Trinity, he was a superior student in both the sciences and the classics. Prior to receiving his degree, because of his brilliance he was appointed the Astronomer Royal of Ireland, beating out several famous astronomers for the post. He held this position until his death, living and working at Dunsink Observatory outside of Dublin. Hamilton made important contributions to optics, abstract algebra, and dynamics. Hamilton invented algebraic objects called quaternions as an example of a noncommutative system. He discovered the appropriate way to multiply quaternions while walking along a canal in Dublin. In his excitement, he carved the formula in the stone of a bridge crossing the canal, a spot marked today by a plaque. Later, Hamilton remained obsessed with quaternions, working to apply them to other areas of mathematics, instead of moving to new areas of research.

In 1857 Hamilton invented "The Icosian Game" based on his work in noncommutative algebra. He sold the idea for 25 pounds to a dealer in games and puzzles. (Because the game never sold well, this turned out to be a bad investment for the dealer.) The "Traveler's Dodecahedron," also called "A Voyage Round the World," the puzzle described in this section, is a variant of that game.

Hamilton married his third love in 1833, but his marriage worked out poorly, because his wife, a semi-invalid, was unable to cope with his household affairs. He suffered from alcoholism and lived reclusively for the last two decades of his life. He died from gout in 1865, leaving masses of papers containing unpublished research. Mixed in with these papers were a large number of dinner plates, many containing the remains of desiccated, uneaten chops.

**EXAMPLE 7** Show that  $K_n$  has a Hamilton circuit whenever  $n \geq 3$ .

*Solution:* We can form a Hamilton circuit in  $K_n$  beginning at any vertex. Such a circuit can be built by visiting vertices in any order we choose, as long as the path begins and ends at the same vertex and visits each other vertex exactly once. This is possible because there are edges in  $K_n$  between any two vertices. ◀

Although no useful necessary and sufficient conditions for the existence of Hamilton circuits are known, quite a few sufficient conditions have been found. Note that the more edges a graph has, the more likely it is to have a Hamilton circuit. Furthermore, adding edges (but not vertices) to a graph with a Hamilton circuit produces a graph with the same Hamilton circuit. So as we add edges to a graph, especially when we make sure to add edges to each vertex, we make it increasingly likely that a Hamilton circuit exists in this graph. Consequently, we would expect there to be sufficient conditions for the existence of Hamilton circuits that depend on the degrees of vertices being sufficiently large. We state two of the most important sufficient conditions here. These conditions were found by Gabriel A. Dirac in 1952 and Oystein Ore in 1960.

**THEOREM 3 DIRAC'S THEOREM** If  $G$  is a simple graph with  $n$  vertices with  $n \geq 3$  such that the degree of every vertex in  $G$  is at least  $n/2$ , then  $G$  has a Hamilton circuit.

**THEOREM 4 ORE'S THEOREM** If  $G$  is a simple graph with  $n$  vertices with  $n \geq 3$  such that  $\deg(u) + \deg(v) \geq n$  for every pair of nonadjacent vertices  $u$  and  $v$  in  $G$ , then  $G$  has a Hamilton circuit.

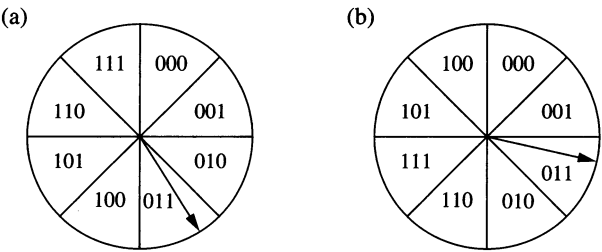
The proof of Ore's Theorem is outlined in Exercise 65 at the end of this section. Dirac's Theorem can be proved as a corollary to Ore's Theorem because the conditions of Dirac's Theorem imply those of Ore's Theorem.

Both Ore's Theorem and Dirac's Theorem provide sufficient conditions for a connected simple graph to have a Hamilton circuit. However, these theorems do not provide necessary conditions for the existence of a Hamilton circuit. For example, the graph  $C_5$  has a Hamilton circuit but does not satisfy the hypotheses of either Ore's Theorem or Dirac's Theorem, as the reader can verify.



The best algorithms known for finding a Hamilton circuit in a graph or determining that no such circuit exists have exponential worst-case time complexity (in the number of vertices of the graph). Finding an algorithm that solves this problem with polynomial worst-case time complexity would be a major accomplishment because it has been shown that this problem is NP-complete (see Section 3.3). Consequently, the existence of such an algorithm would imply that many other seemingly intractable problems could be solved using algorithms with polynomial worst-case time complexity.

Hamilton paths and circuits can be used to solve practical problems. For example, many applications ask for a path or circuit that visits each road intersection in a city, each place pipelines intersect in a utility grid, or each node in a communications network exactly once. Finding a Hamilton path or circuit in the appropriate graph model can solve such problems. The famous **traveling salesman problem** asks for the shortest route a traveling salesman should take to visit a set of cities. This problem reduces to finding a Hamilton circuit in a complete graph such that the total weight of its edges is as small as possible. We will return to this question in Section 9.6.



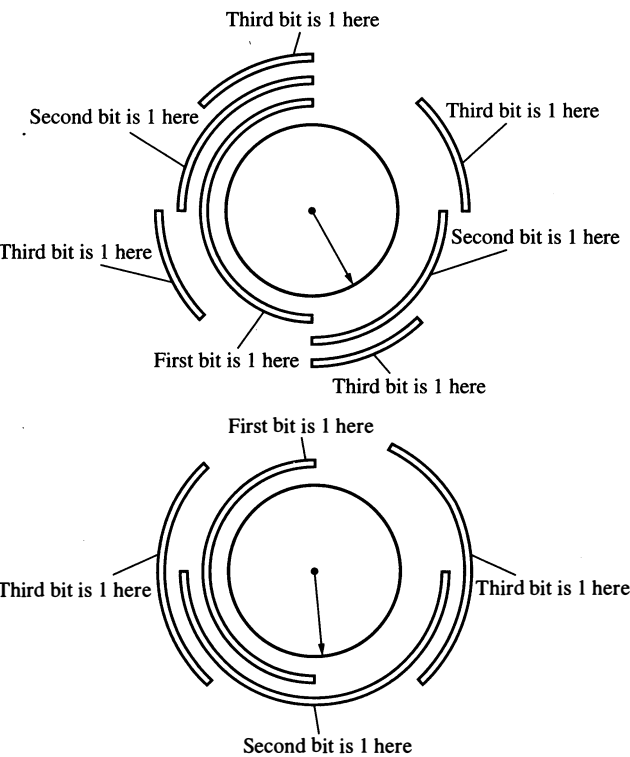
**FIGURE 12** Converting the Position of a Pointer into Digital Form.

We now describe a less obvious application of Hamilton circuits to coding.

**EXAMPLE 8 Gray Codes** The position of a rotating pointer can be represented in digital form. One way to do this is to split the circle into  $2^n$  arcs of equal length and to assign a bit string of length  $n$  to each arc. Two ways to do this using bit strings of length three are shown in Figure 12.

The digital representation of the position of the pointer can be determined using a set of  $n$  contacts. Each contact is used to read one bit in the digital representation of the position. This is illustrated in Figure 13 for the two assignments from Figure 12.

When the pointer is near the boundary of two arcs, a mistake may be made in reading its position. This may result in a major error in the bit string read. For instance, in the coding



**FIGURE 13** The Digital Representation of the Position of the Pointer.

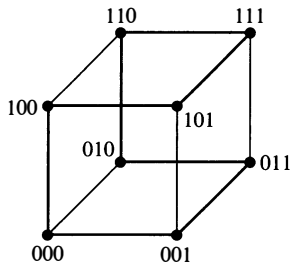


FIGURE 14 A Hamilton Circuit for  $Q_3$ .

scheme in Figure 12(a), if a small error is made in determining the position of the pointer, the bit string 100 is read instead of 011. All three bits are incorrect! To minimize the effect of an error in determining the position of the pointer, the assignment of the bit strings to the  $2^n$  arcs should be made so that only one bit is different in the bit strings represented by adjacent arcs. This is exactly the situation in the coding scheme in Figure 12(b). An error in determining the position of the pointer gives the bit string 010 instead of 011. Only one bit is wrong.

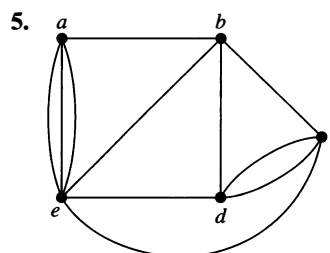
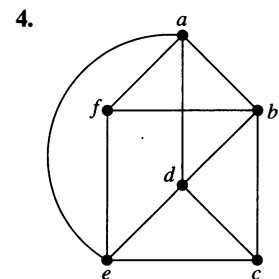
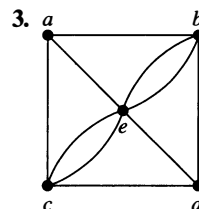
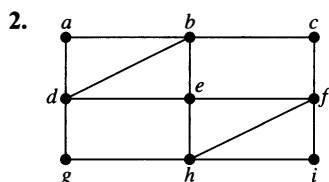
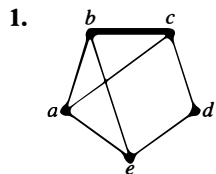


A **Gray code** is a labeling of the arcs of the circle such that adjacent arcs are labeled with bit strings that differ in exactly one bit. The assignment in Figure 12(b) is a Gray code. We can find a Gray code by listing all bit strings of length  $n$  in such a way that each string differs in exactly one position from the preceding bit string, and the last string differs from the first in exactly one position. We can model this problem using the  $n$ -cube  $Q_n$ . What is needed to solve this problem is a Hamilton circuit in  $Q_n$ . Such Hamilton circuits are easily found. For instance, a Hamilton circuit for  $Q_3$  is displayed in Figure 14. The sequence of bit strings differing in exactly one bit produced by this Hamilton circuit is 000, 001, 011, 010, 110, 111, 101, 100.

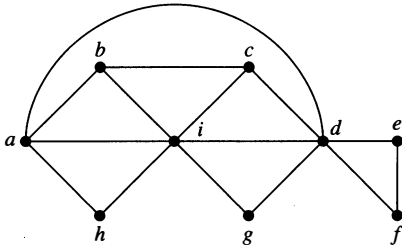
Gray codes are named after Frank Gray, who invented them in the 1940s at AT&T Bell Laboratories to minimize the effect of errors in transmitting digital signals. ◀

## Exercises

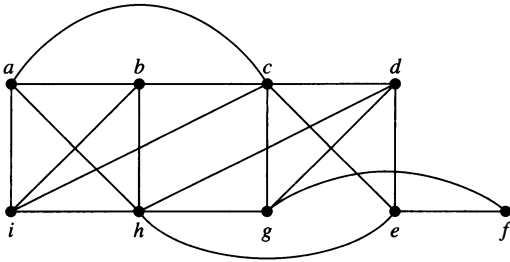
In Exercises 1–8 determine whether the given graph has an Euler circuit. Construct such a circuit when one exists. If no Euler circuit exists, determine whether the graph has an Euler path and construct such a path if one exists.



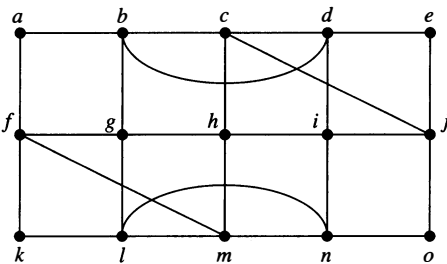
6.



7.

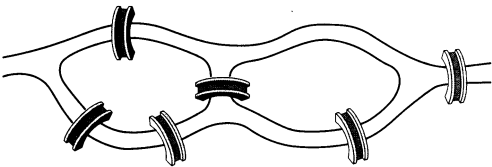


8.



9. In Kaliningrad (the Russian name for Königsberg) there are two additional bridges, besides the seven that were present in the eighteenth century. These new bridges connect regions  $B$  and  $C$  and regions  $B$  and  $D$ , respectively. Can someone cross all nine bridges in Kaliningrad exactly once and return to the starting point?

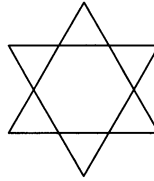
10. Can someone cross all the bridges shown in this map exactly once and return to the starting point?



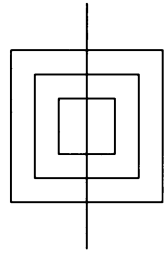
11. When can the centerlines of the streets in a city be painted without traveling a street more than once? (Assume that all the streets are two-way streets.)
12. Devise a procedure, similar to Algorithm 1, for constructing Euler paths in multigraphs.

In Exercises 13–15 determine whether the picture shown can be drawn with a pencil in a continuous motion without lifting the pencil or retracing part of the picture.

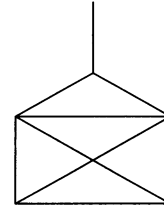
13.



14.



15.

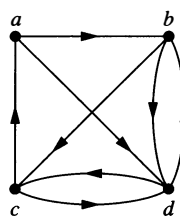


- \*16. Show that a directed multigraph having no isolated vertices has an Euler circuit if and only if the graph is weakly connected and the in-degree and out-degree of each vertex are equal.

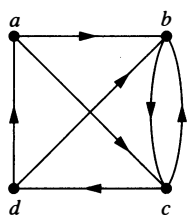
- \*17. Show that a directed multigraph having no isolated vertices has an Euler path but not an Euler circuit if and only if the graph is weakly connected and the in-degree and out-degree of each vertex are equal for all but two vertices, one that has in-degree one larger than its out-degree and the other that has out-degree one larger than its in-degree.

In Exercises 18–23 determine whether the directed graph shown has an Euler circuit. Construct an Euler circuit if one exists. If no Euler circuit exists, determine whether the directed graph has an Euler path. Construct an Euler path if one exists.

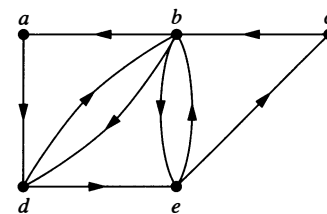
18.



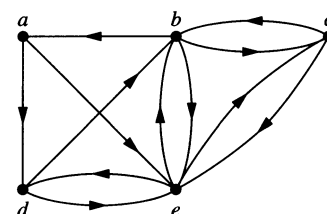
19.



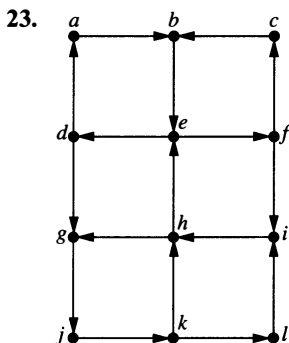
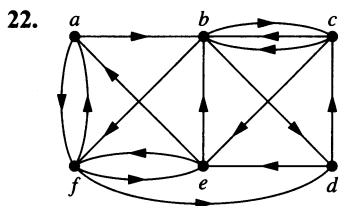
20.



21.







\*24. Devise an algorithm for constructing Euler circuits in directed graphs.

25. Devise an algorithm for constructing Euler paths in directed graphs.

26. For which values of  $n$  do these graphs have an Euler circuit?

- a)  $K_n$    b)  $C_n$    c)  $W_n$    d)  $Q_n$

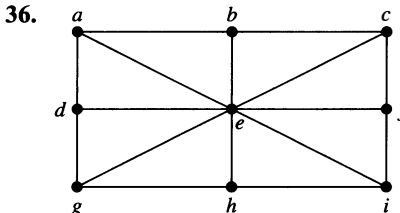
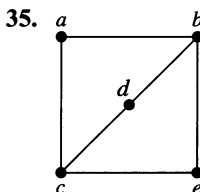
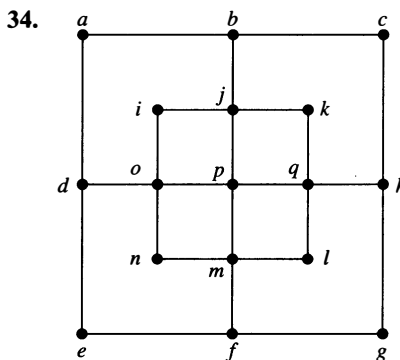
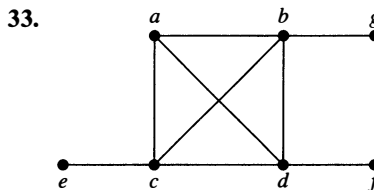
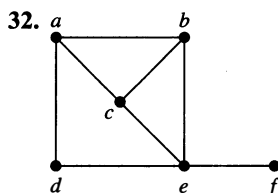
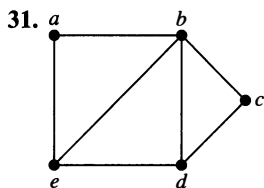
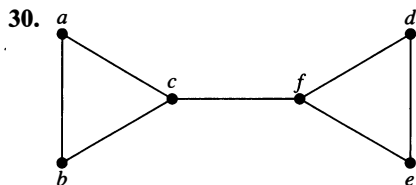
27. For which values of  $n$  do the graphs in Exercise 26 have an Euler path but no Euler circuit?

28. For which values of  $m$  and  $n$  does the complete bipartite graph  $K_{m,n}$  have an

- a) Euler circuit?  
b) Euler path?

29. Find the least number of times it is necessary to lift a pencil from the paper when drawing each of the graphs in Exercises 1–7 without retracing any part of the graph.

In Exercises 30–36 determine whether the given graph has a Hamilton circuit. If it does, find such a circuit. If it does not, give an argument to show why no such circuit exists.



37. Does the graph in Exercise 30 have a Hamilton path? If so, find such a path. If it does not, give an argument to show why no such path exists.

38. Does the graph in Exercise 31 have a Hamilton path? If so, find such a path. If it does not, give an argument to show why no such path exists.

39. Does the graph in Exercise 32 have a Hamilton path? If so, find such a path. If it does not, give an argument to show why no such path exists.

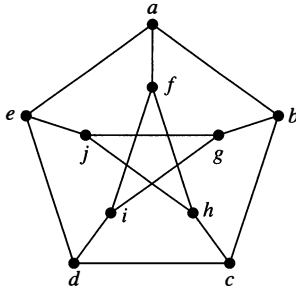
40. Does the graph in Exercise 33 have a Hamilton path? If so, find such a path. If it does not, give an argument to show why no such path exists.

\*41. Does the graph in Exercise 34 have a Hamilton path? If so, find such a path. If it does not, give an argument to show why no such path exists.

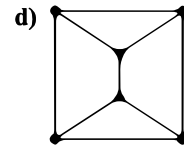
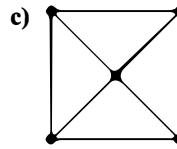
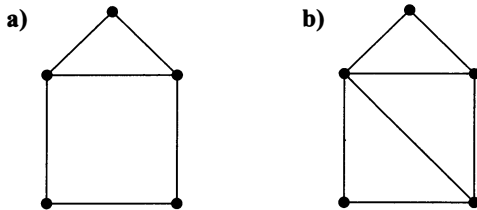
42. Does the graph in Exercise 35 have a Hamilton path? If so, find such a path. If it does not, give an argument to show why no such path exists.

43. Does the graph in Exercise 36 have a Hamilton path? If so, find such a path. If it does not, give an argument to show why no such path exists.

44. For which values of  $n$  do the graphs in Exercise 26 have a Hamilton circuit?
45. For which values of  $m$  and  $n$  does the complete bipartite graph  $K_{m,n}$  have a Hamilton circuit?
- \*46. Show that the **Petersen graph**, shown here, does not have a Hamilton circuit, but that the subgraph obtained by deleting a vertex  $v$ , and all edges incident with  $v$ , does have a Hamilton circuit.



47. For each of these graphs, determine (i) whether Dirac's Theorem can be used to show that the graph has a Hamilton circuit, (ii) whether Ore's Theorem can be used to show that the graph has a Hamilton circuit, and (iii) whether the graph has a Hamilton circuit.



48. Can you find a simple graph with  $n$  vertices with  $n \geq 3$  that does not have a Hamilton circuit, yet the degree of every vertex in the graph is at least  $(n-1)/2$ ?

- \*49. Show that there is a Gray code of order  $n$  whenever  $n$  is a positive integer, or equivalently, show that the  $n$ -cube  $Q_n$ ,  $n > 1$ , always has a Hamilton circuit. [Hint: Use mathematical induction. Show how to produce a Gray code of order  $n$  from one of order  $n-1$ .]



**Fleury's algorithm** for constructing Euler circuits begins with an arbitrary vertex of a connected multigraph and forms a circuit by choosing edges successively. Once an edge is chosen, it is removed. Edges are chosen successively so that each edge begins where the last edge ends, and so that this edge is not a cut edge unless there is no alternative.

50. Use Fleury's algorithm to find an Euler circuit in the graph  $G$  in Figure 5.

- \*51. Express Fleury's algorithm in pseudocode.

- \*52. Prove that Fleury's algorithm always produces an Euler circuit.

- \*53. Give a variant of Fleury's algorithm to produce Euler paths.

54. A diagnostic message can be sent out over a computer network to perform tests over all links and in all devices. What sort of paths should be used to test all links? To test all devices?

55. Show that a bipartite graph with an odd number of vertices does not have a Hamilton circuit.



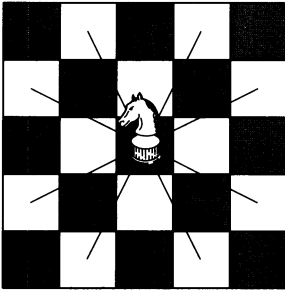
**JULIUS PETER CHRISTIAN PETERSEN (1839–1910)** Julius Petersen was born in the Danish town of Sorø. His father was a dyer. In 1854 his parents were no longer able to pay for his schooling, so he became an apprentice in an uncle's grocery store. When this uncle died, he left Petersen enough money to return to school. After graduating, he began studying engineering at the Polytechnical School in Copenhagen, later deciding to concentrate on mathematics. He published his first textbook, a book on logarithms, in 1858. When his inheritance ran out, he had to teach to make a living. From 1859 until 1871 Petersen taught at a prestigious private high school in Copenhagen. While teaching high school he continued his studies, entering Copenhagen University in 1862. He married Laura Bertelsen in 1862; they had three children, two sons and a daughter.

Petersen obtained a mathematics degree from Copenhagen University in 1866 and finally obtained his doctorate in 1871 from that school. After receiving his doctorate, he taught at a polytechnic and military academy. In 1887 he was appointed to a professorship at the University of Copenhagen. Petersen was well known in Denmark as the author of a large series of textbooks for high schools and universities. One of his books, *Methods and Theories for the Solution of Problems of Geometrical Construction*, was translated into eight languages, with the English language version last reprinted in 1960 and the French version reprinted as recently as 1990, more than a century after the original publication date.

Petersen worked in a wide range of areas, including algebra, analysis, cryptography, geometry, mechanics, mathematical economics, and number theory. His contributions to graph theory, including results on regular graphs, are his best-known work. He was noted for his clarity of exposition, problem-solving skills, originality, sense of humor, vigor, and teaching. One interesting fact about Petersen was that he preferred not to read the writings of other mathematicians. This led him often to rediscover results already proved by others, often with embarrassing consequences. However, he was often angry when other mathematicians did not read his writings!

Petersen's death was front-page news in Copenhagen. A newspaper of the time described him as the Hans Christian Andersen of science—a child of the people who made good in the academic world.

A **knight** is a chess piece that can move either two spaces horizontally and one space vertically or one space horizontally and two spaces vertically. That is, a knight on square  $(x, y)$  can move to any of the eight squares  $(x \pm 2, y \pm 1)$ ,  $(x \pm 1, y \pm 2)$ , if these squares are on the chessboard, as illustrated here.



A **knight's tour** is a sequence of legal moves by a knight starting at some square and visiting each square exactly once. A knight's tour is called **reentrant** if there is a legal move that takes the knight from the last square of the tour back to where the tour began. We can model knight's tours using the graph that has a vertex for each square on the board, with an edge connecting two vertices if a knight can legally move between the squares represented by these vertices.

56. Draw the graph that represents the legal moves of a knight on a  $3 \times 3$  chessboard.
57. Draw the graph that represents the legal moves of a knight on a  $3 \times 4$  chessboard.
58. a) Show that finding a knight's tour on an  $m \times n$  chessboard is equivalent to finding a Hamilton path on the graph representing the legal moves of a knight on that board.  
b) Show that finding a reentrant knight's tour on an  $m \times n$  chessboard is equivalent to finding a Hamilton circuit on the corresponding graph.
- \*59. Show that there is a knight's tour on a  $3 \times 4$  chessboard.
- \*60. Show that there is no knight's tour on a  $3 \times 3$  chessboard.

- \*61. Show that there is no knight's tour on a  $4 \times 4$  chessboard.
62. Show that the graph representing the legal moves of a knight on an  $m \times n$  chessboard, whenever  $m$  and  $n$  are positive integers, is bipartite.
63. Show that there is no reentrant knight's tour on an  $m \times n$  chessboard when  $m$  and  $n$  are both odd. [Hint: Use Exercises 55, 58b, and 62.]
- \*64. Show that there is a knight's tour on an  $8 \times 8$  chessboard. [Hint: You can construct a knight's tour using a method invented by H. C. Warnsdorff in 1823: Start in any square, and then always move to a square connected to the fewest number of unused squares. Although this method may not always produce a knight's tour, it often does.]
65. The parts of this exercise outline a proof of Ore's Theorem. Suppose that  $G$  is a simple graph with  $n$  vertices,  $n \geq 3$ , and  $\deg(x) + \deg(y) \geq n$  whenever  $x$  and  $y$  are nonadjacent vertices in  $G$ . Ore's Theorem states that under these conditions,  $G$  has a Hamilton circuit.
  - a) Show that if  $G$  does not have a Hamilton circuit, then there exists another graph  $H$  with the same vertices as  $G$ , which can be constructed by adding edges to  $G$  such that the addition of a single edge would produce a Hamilton circuit in  $H$ . [Hint: Add as many edges as possible at each successive vertex of  $G$  without producing a Hamilton circuit.]
  - b) Show that there is a Hamilton path in  $H$ .
  - c) Let  $v_1, v_2, \dots, v_n$  be a Hamilton path in  $H$ . Show that  $\deg(v_1) + \deg(v_n) \geq n$  and that there are at most  $\deg(v_1)$  vertices not adjacent to  $v_n$  (including  $v_n$  itself).
  - d) Let  $S$  be the set of vertices preceding each vertex adjacent to  $v_1$  in the Hamilton path. Show that  $S$  contains  $\deg(v_1)$  vertices and  $v_n \notin S$ .
  - e) Show that  $S$  contains a vertex  $v_k$ , which is adjacent to  $v_n$ , implying that there are edges connecting  $v_1$  and  $v_{k+1}$  and  $v_k$  and  $v_n$ .
  - f) Show that part (e) implies that  $v_1, v_2, \dots, v_{k-1}, v_k, v_n, v_{n-1}, \dots, v_{k+1}, v_1$  is a Hamilton circuit in  $G$ . Conclude from this contradiction that Ore's Theorem holds.

## 9.6 Shortest-Path Problems

### Introduction

Many problems can be modeled using graphs with weights assigned to their edges. As an illustration, consider how an airline system can be modeled. We set up the basic graph model by representing cities by vertices and flights by edges. Problems involving distances can be modeled by assigning distances between cities to the edges. Problems involving flight time can be modeled by assigning flight times to edges. Problems involving fares can be modeled by assigning fares to the edges. Figure 1 displays three different assignments of weights to the edges of a graph representing distances, flight times, and fares, respectively.

Graphs that have a number assigned to each edge are called **weighted graphs**. Weighted graphs are used to model computer networks. Communications costs (such as the monthly cost

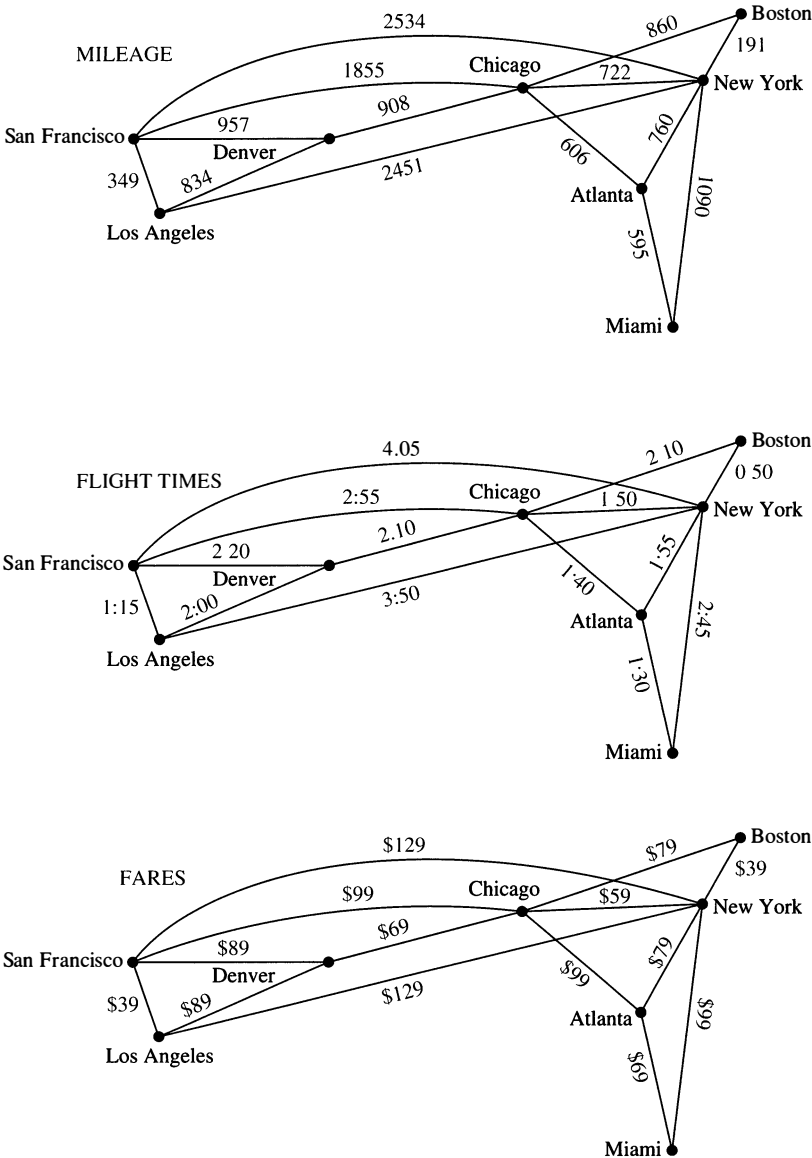
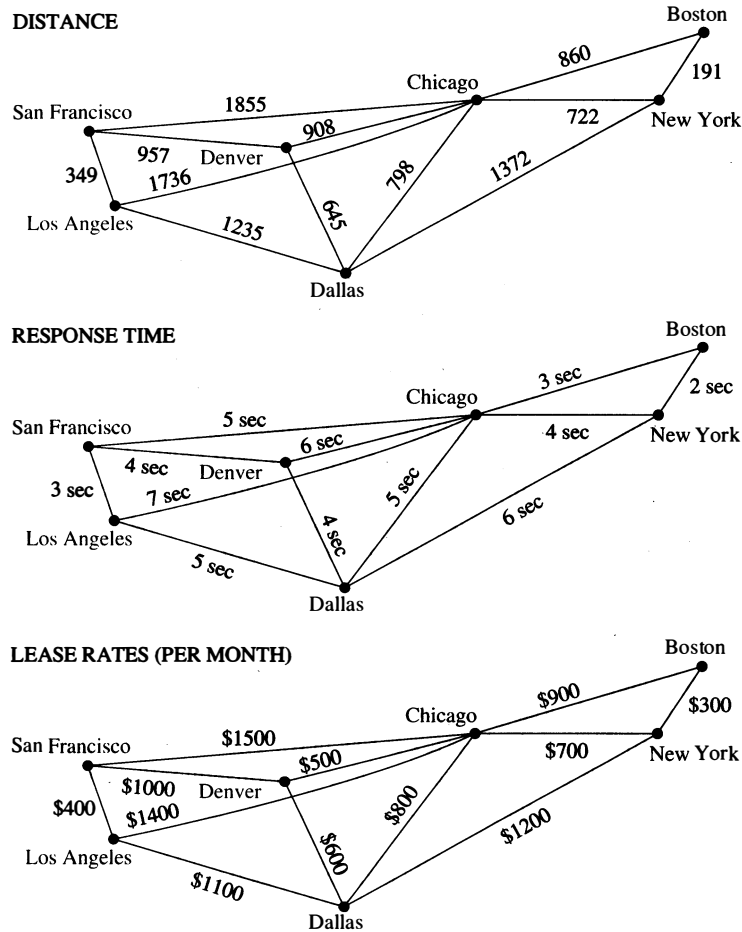


FIGURE 1 Weighted Graphs Modeling an Airline System.

of leasing a telephone line), the response times of the computers over these lines, or the distance between computers, can all be studied using weighted graphs. Figure 2 displays weighted graphs that represent three ways to assign weights to the edges of a graph of a computer network, corresponding to distance, response time, and cost.

Several types of problems involving weighted graphs arise frequently. Determining a path of least length between two vertices in a network is one such problem. To be more specific, let the **length** of a path in a weighted graph be the sum of the weights of the edges of this path. (The reader should note that this use of the term *length* is different from the use of *length* to denote the number of edges in a path in a graph without weights.) The question is: What is a shortest path, that is, a path of least length, between two given vertices? For instance, in the airline system represented by the weighted graph shown in Figure 1, what is a shortest path in air distance between Boston and Los Angeles? What combinations of flights has the smallest total



**FIGURE 2** Weighted Graphs Modeling a Computer Network.

flight time (that is, total time in the air, not including time between flights) between Boston and Los Angeles? What is the cheapest fare between these two cities? In the computer network shown in Figure 2, what is a least expensive set of telephone lines needed to connect the computers in San Francisco with those in New York? Which set of telephone lines gives a fastest response time for communications between San Francisco and New York? Which set of lines has a shortest overall distance?

Another important problem involving weighted graphs asks for a circuit of shortest total length that visits every vertex of a complete graph exactly once. This is the famous *traveling salesman problem*, which asks for an order in which a salesman should visit each of the cities on his route exactly once so that he travels the minimum total distance. We will discuss the traveling salesman problem later in this section.

## A Shortest-Path Algorithm



There are several different algorithms that find a shortest path between two vertices in a weighted graph. We will present an algorithm discovered by the Dutch mathematician Edsger Dijkstra in 1959. The version we will describe solves this problem in undirected weighted graphs where all the weights are positive. It is easy to adapt it to solve shortest-path problems in directed graphs.

Before giving a formal presentation of the algorithm, we will give a motivating example.

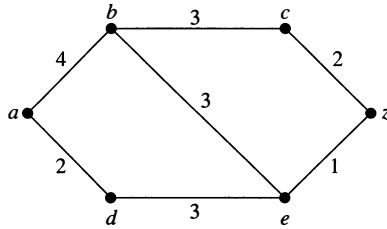


FIGURE 3 A Weighted Simple Graph.

**EXAMPLE 1** What is the length of a shortest path between  $a$  and  $z$  in the weighted graph shown in Figure 3?

*Solution:* Although a shortest path is easily found by inspection, we will develop some ideas useful in understanding Dijkstra's algorithm. We will solve this problem by finding the length of a shortest path from  $a$  to successive vertices, until  $z$  is reached.

The only paths starting at  $a$  that contain no vertex other than  $a$  (until the terminal vertex is reached) are  $a, b$  and  $a, d$ . Because the lengths of  $a, b$  and  $a, d$  are 4 and 2, respectively, it follows that  $d$  is the closest vertex to  $a$ .

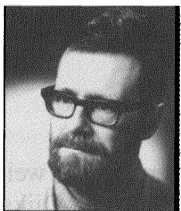
We can find the next closest vertex by looking at all paths that go through only  $a$  and  $d$  (until the terminal vertex is reached). The shortest such path to  $b$  is still  $a, b$ , with length 4, and the shortest such path to  $e$  is  $a, d, e$ , with length 5. Consequently, the next closest vertex to  $a$  is  $b$ .

To find the third closest vertex to  $a$ , we need to examine only paths that go through only  $a, d$ , and  $b$  (until the terminal vertex is reached). There is a path of length 7 to  $c$ , namely,  $a, b, c$ , and a path of length 6 to  $z$ , namely,  $a, d, e, z$ . Consequently,  $z$  is the next closest vertex to  $a$ , and the length of a shortest path to  $z$  is 6. ◀

Example 1 illustrates the general principles used in Dijkstra's algorithm. Note that a shortest path from  $a$  to  $z$  could have been found by inspection. However, inspection is impractical for both humans and computers for graphs with large numbers of edges.

We will now consider the general problem of finding the length of a shortest path between  $a$  and  $z$  in an undirected connected simple weighted graph. Dijkstra's algorithm proceeds by finding the length of a shortest path from  $a$  to a first vertex, the length of a shortest path from  $a$  to a second vertex, and so on, until the length of a shortest path from  $a$  to  $z$  is found.

The algorithm relies on a series of iterations. A distinguished set of vertices is constructed by adding one vertex at each iteration. A labeling procedure is carried out at each iteration. In this labeling procedure, a vertex  $w$  is labeled with the length of a shortest path from  $a$  to  $w$  that contains only vertices already in the distinguished set. The vertex added to the distinguished set is one with a minimal label among those vertices not already in the set.



**EDSGER WYBE DIJKSTRA (1930–2002)** Edsger Dijkstra, born in the Netherlands, began programming computers in the early 1950s while studying theoretical physics at the University of Leiden. In 1952, realizing that he was more interested in programming than in physics, he quickly completed the requirements for his physics degree and began his career as a programmer, even though programming was not a recognized profession. (In 1957, the authorities in Amsterdam refused to accept “programming” as his profession on his marriage license. However, they did accept “theoretical physicist” when he changed his entry to this.)

Dijkstra was one of the most forceful proponents of programming as a scientific discipline. He has made fundamental contributions to the areas of operating systems, including deadlock avoidance; programming languages, including the notion of structured programming; and algorithms. In 1972 Dijkstra received the Turing

Award from the Association for Computing Machinery, one of the most prestigious awards in computer science. Dijkstra became a Burroughs Research Fellow in 1973, and in 1984 he was appointed to a chair in Computer Science at the University of Texas,

We now give the details of Dijkstra's algorithm. It begins by labeling  $a$  with 0 and the other vertices with  $\infty$ . We use the notation  $L_0(a) = 0$  and  $L_0(v) = \infty$  for these labels before any iterations have taken place (the subscript 0 stands for the "0th" iteration). These labels are the lengths of shortest paths from  $a$  to the vertices, where the paths contain only the vertex  $a$ . (Because no path from  $a$  to a vertex different from  $a$  exists,  $\infty$  is the length of a shortest path between  $a$  and this vertex.)

Dijkstra's algorithm proceeds by forming a distinguished set of vertices. Let  $S_k$  denote this set after  $k$  iterations of the labeling procedure. We begin with  $S_0 = \emptyset$ . The set  $S_k$  is formed from  $S_{k-1}$  by adding a vertex  $u$  not in  $S_{k-1}$  with the smallest label. Once  $u$  is added to  $S_k$ , we update the labels of all vertices not in  $S_k$ , so that  $L_k(v)$ , the label of the vertex  $v$  at the  $k$ th stage, is the length of a shortest path from  $a$  to  $v$  that contains vertices only in  $S_k$  (that is, vertices that were already in the distinguished set together with  $u$ ).

Let  $v$  be a vertex not in  $S_k$ . To update the label of  $v$ , note that  $L_k(v)$  is the length of a shortest path from  $a$  to  $v$  containing only vertices in  $S_k$ . The updating can be carried out efficiently when this observation is used: A shortest path from  $a$  to  $v$  containing only elements of  $S_k$  is either a shortest path from  $a$  to  $v$  that contains only elements of  $S_{k-1}$  (that is, the distinguished vertices not including  $u$ ), or it is a shortest path from  $a$  to  $u$  at the  $(k-1)$ st stage with the edge  $(u, v)$  added. In other words,

$$L_k(a, v) = \min\{L_{k-1}(a, v), L_{k-1}(a, u) + w(u, v)\}.$$

This procedure is iterated by successively adding vertices to the distinguished set until  $z$  is added. When  $z$  is added to the distinguished set, its label is the length of a shortest path from  $a$  to  $z$ . Dijkstra's algorithm is given in Algorithm 1. Later we will give a proof that this algorithm is correct.

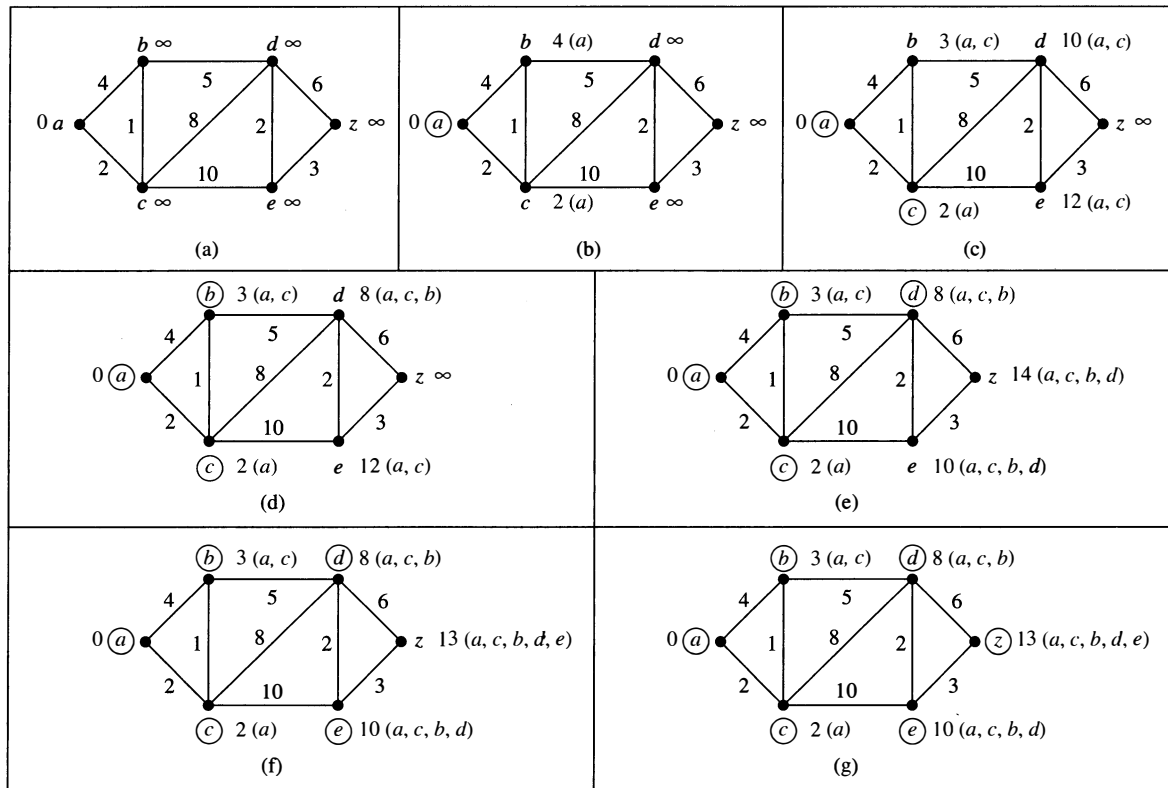
#### ALGORITHM 1 Dijkstra's Algorithm.

```

procedure Dijkstra( $G$ : weighted connected simple graph, with
    all weights positive)
  { $G$  has vertices  $a = v_0, v_1, \dots, v_n = z$  and weights  $w(v_i, v_j)$ 
    where  $w(v_i, v_j) = \infty$  if  $\{v_i, v_j\}$  is not an edge in  $G$ }
  for  $i := 1$  to  $n$ 
     $L(v_i) := \infty$ 
   $L(a) := 0$ 
   $S := \emptyset$ 
  {the labels are now initialized so that the label of  $a$  is 0 and all
    other labels are  $\infty$ , and  $S$  is the empty set}
  while  $z \notin S$ 
    begin
       $u :=$  a vertex not in  $S$  with  $L(u)$  minimal
       $S := S \cup \{u\}$ 
      for all vertices  $v$  not in  $S$ 
        if  $L(u) + w(u, v) < L(v)$  then  $L(v) := L(u) + w(u, v)$ 
        {this adds a vertex to  $S$  with minimal label and updates the
          labels of vertices not in  $S$ }
      end { $L(z)$  = length of a shortest path from  $a$  to  $z$ }

```

Example 2 illustrates how Dijkstra's algorithm works. Afterward, we will show that this algorithm always produces the length of a shortest path between two vertices in a weighted graph.



**FIGURE 4** Using Dijkstra's Algorithm to Find a Shortest Path from  $a$  to  $z$ .

**EXAMPLE 2** Use Dijkstra's algorithm to find the length of a shortest path between the vertices  $a$  and  $z$  in the weighted graph displayed in Figure 4(a).

**Solution:** The steps used by Dijkstra's algorithm to find a shortest path between  $a$  and  $z$  are shown in Figure 4. At each iteration of the algorithm the vertices of the set  $S_k$  are circled. A shortest path from  $a$  to each vertex containing only vertices in  $S_k$  is indicated for each iteration. The algorithm terminates when  $z$  is circled. We find that a shortest path from  $a$  to  $z$  is  $a, c, b, d, e, z$ , with length 13. ◀

**Remark:** In performing Dijkstra's algorithm it is sometimes more convenient to keep track of labels of vertices in each step using a table instead of redrawing the graph for each step.

Next, we use an inductive argument to show that Dijkstra's algorithm produces the length of a shortest path between two vertices  $a$  and  $z$  in an undirected connected weighted graph. Take as the induction hypothesis the following assertion: At the  $k$ th iteration

- (i) the label of every vertex  $v$  in  $S$  is the length of a shortest path from  $a$  to this vertex, and
- (ii) the label of every vertex not in  $S$  is the length of a shortest path from  $a$  to this vertex that contains only (besides the vertex itself) vertices in  $S$ .

When  $k = 0$ , before any iterations are carried out,  $S = \emptyset$ , so the length of a shortest path from  $a$  to a vertex other than  $a$  is  $\infty$ . Hence, the basis case is true.

Assume that the inductive hypothesis holds for the  $k$ th iteration. Let  $v$  be the vertex added to  $S$  at the  $(k + 1)$ st iteration, so  $v$  is a vertex not in  $S$  at the end of the  $k$ th iteration with the smallest label (in the case of ties, any vertex with smallest label may be used).



From the inductive hypothesis we see that the vertices in  $S$  before the  $(k + 1)$ st iteration are labeled with the length of a shortest path from  $a$ . Also,  $v$  must be labeled with the length of a shortest path to it from  $a$ . If this were not the case, at the end of the  $k$ th iteration there would be a path of length less than  $L_k(v)$  containing a vertex not in  $S$  [because  $L_k(v)$  is the length of a shortest path from  $a$  to  $v$  containing only vertices in  $S$  after the  $k$ th iteration]. Let  $u$  be the first vertex not in  $S$  in such a path. There is a path with length less than  $L_k(v)$  from  $a$  to  $u$  containing only vertices of  $S$ . This contradicts the choice of  $v$ . Hence, (i) holds at the end of the  $(k + 1)$ st iteration.

Let  $u$  be a vertex not in  $S$  after  $k + 1$  iterations. A shortest path from  $a$  to  $u$  containing only elements of  $S$  either contains  $v$  or it does not. If it does not contain  $v$ , then by the inductive hypothesis its length is  $L_k(u)$ . If it does contain  $v$ , then it must be made up of a path from  $a$  to  $v$  of shortest possible length containing elements of  $S$  other than  $v$ , followed by the edge from  $v$  to  $u$ . In this case, its length would be  $L_k(v) + w(v, u)$ . This shows that (ii) is true, because  $L_{k+1}(u) = \min\{L_k(u), L_k(v) + w(v, u)\}$ .

Theorem 1 has been proved.

**THEOREM 1** Dijkstra's algorithm finds the length of a shortest path between two vertices in a connected simple undirected weighted graph.

We can now estimate the computational complexity of Dijkstra's algorithm (in terms of additions and comparisons). The algorithm uses no more than  $n - 1$  iterations, because one vertex is added to the distinguished set at each iteration. We are done if we can estimate the number of operations used for each iteration. We can identify the vertex not in  $S_k$  with the smallest label using no more than  $n - 1$  comparisons. Then we use an addition and a comparison to update the label of each vertex not in  $S_k$ . It follows that no more than  $2(n - 1)$  operations are used at each iteration, because there are no more than  $n - 1$  labels to update at each iteration. Because we use no more than  $n - 1$  iterations, each using no more than  $2(n - 1)$  operations, we have Theorem 2.

**THEOREM 2** Dijkstra's algorithm uses  $O(n^2)$  operations (additions and comparisons) to find the length of a shortest path between two vertices in a connected simple undirected weighted graph with  $n$  vertices.

## The Traveling Salesman Problem



We now discuss an important problem involving weighted graphs. Consider the following problem: A traveling salesman wants to visit each of  $n$  cities exactly once and return to his starting point. For example, suppose that the salesman wants to visit Detroit, Toledo, Saginaw, Grand Rapids, and Kalamazoo (see Figure 5). In which order should he visit these cities to travel the minimum total distance? To solve this problem we can assume the salesman starts in Detroit (because this must be part of the circuit) and examine all possible ways for him to visit the other four cities and then return to Detroit (starting elsewhere will produce the same circuits). There are a total of 24 such circuits, but because we travel the same distance when we travel a circuit in reverse order, we need only consider 12 different circuits to find the minimum total distance he must travel. We list these 12 different circuits and the total distance traveled for each circuit. As can be seen from the list, the minimum total distance of 458 miles is traveled using the circuit Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit (or its reverse).

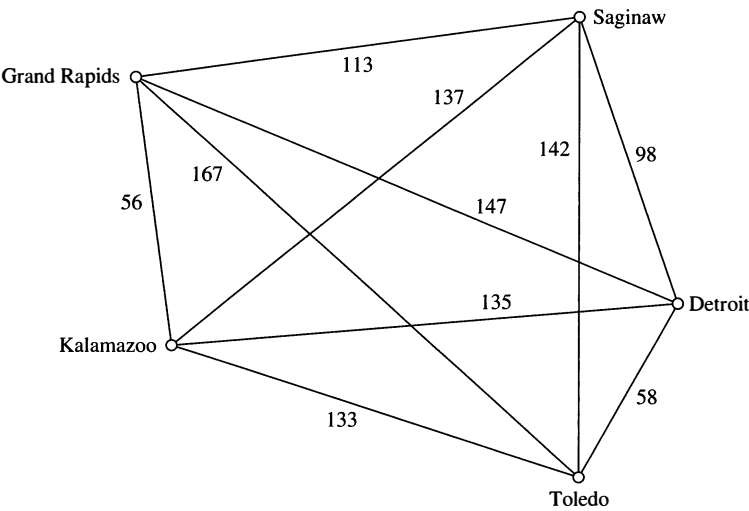


FIGURE 5 The Graph Showing the Distances between Five Cities.

Route	Total Distance (miles)
Detroit–Toledo–Grand Rapids–Saginaw–Kalamazoo–Detroit	610
Detroit–Toledo–Grand Rapids–Kalamazoo–Saginaw–Detroit	516
Detroit–Toledo–Kalamazoo–Saginaw–Grand Rapids–Detroit	588
Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit	458
Detroit–Toledo–Saginaw–Kalamazoo–Grand Rapids–Detroit	540
Detroit–Toledo–Saginaw–Grand Rapids–Kalamazoo–Detroit	504
Detroit–Saginaw–Toledo–Grand Rapids–Kalamazoo–Detroit	598
Detroit–Saginaw–Toledo–Kalamazoo–Grand Rapids–Detroit	576
Detroit–Saginaw–Kalamazoo–Toledo–Grand Rapids–Detroit	682
Detroit–Saginaw–Grand Rapids–Toledo–Kalamazoo–Detroit	646
Detroit–Grand Rapids–Saginaw–Toledo–Kalamazoo–Detroit	670
Detroit–Grand Rapids–Toledo–Saginaw–Kalamazoo–Detroit	728

We just described an instance of the **traveling salesman problem**. The traveling salesman problem asks for the circuit of minimum total weight in a weighted, complete, undirected graph that visits each vertex exactly once and returns to its starting point. This is equivalent to asking for a Hamilton circuit with minimum total weight in the complete graph, because each vertex is visited exactly once in the circuit.

The most straightforward way to solve an instance of the traveling salesman problem is to examine all possible Hamilton circuits and select one of minimum total length. How many circuits do we have to examine to solve the problem if there are  $n$  vertices in the graph? Once a starting point is chosen, there are  $(n - 1)!$  different Hamilton circuits to examine, because there are  $n - 1$  choices for the second vertex,  $n - 2$  choices for the third vertex, and so on. Because a Hamilton circuit can be traveled in reverse order, we need only examine  $(n - 1)!/2$  circuits to find our answer. Note that  $(n - 1)!/2$  grows extremely rapidly. Trying to solve a traveling salesman problem in this way when there are only a few dozen vertices is impractical. For example, with 25 vertices, a total of  $24!/2$  (approximately  $3.1 \times 10^{23}$ ) different Hamilton circuits would have to be considered. If it took just one nanosecond ( $10^{-9}$  second) to examine each Hamilton circuit, a total of approximately ten million years would be required to find a minimum-length Hamilton circuit in this graph by exhaustive search techniques.

Because the traveling salesman problem has both practical and theoretical importance, a great deal of effort has been devoted to devising efficient algorithms that solve it. However, no algorithm with polynomial worst-case time complexity is known for solving this problem. Furthermore, if a polynomial worst-case time complexity algorithm were discovered for the traveling salesman problem, many other difficult problems would also be solvable using polynomial worst-case time complexity algorithms (such as determining whether a proposition in  $n$  variables is a tautology, discussed in Chapter 1). This follows from the theory of NP-completeness. (For more information about this, consult [GaJo79].)

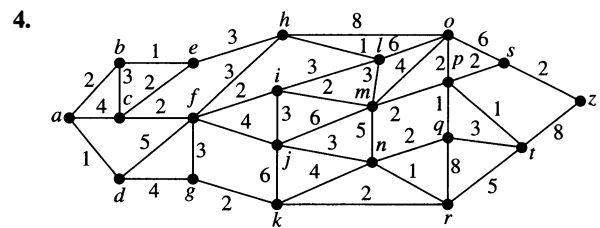
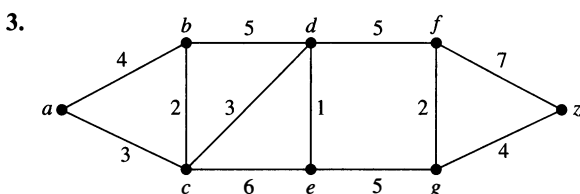
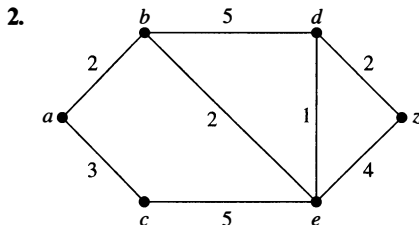
A practical approach to the traveling salesman problem when there are many vertices to visit is to use an **approximation algorithm**. These are algorithms that do not necessarily produce the exact solution to the problem but instead are guaranteed to produce a solution that is close to an exact solution. That is, they may produce a Hamilton circuit with total weight  $W'$  such that  $W \leq W' \leq cW$ , where  $W$  is the total length of an exact solution and  $c$  is a constant. For example, there is an algorithm with polynomial worst-case time complexity that works if the weighted graph satisfies the triangle inequality such that  $c = 3/2$ . For general weighted graphs for every positive real number  $k$  no algorithm is known that will always produce a solution at most  $k$  times a best solution. If such an algorithm existed, this would show that the class P would be the same as the class NP, perhaps the most famous open question about the complexity of algorithms (see Section 3.3).

In practice, algorithms have been developed that can solve traveling salesman problems with as many as 1000 vertices within 2% of an exact solution using only a few minutes of computer time. For more information about the traveling salesman problem, including history, applications, and algorithms, see the chapter on this topic in *Applications of Discrete Mathematics* [MiRo91] also available on the website for this book.

## Exercises

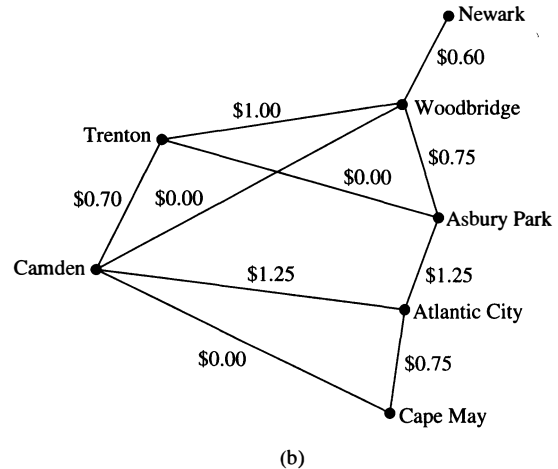
- For each of these problems about a subway system, describe a weighted graph model that can be used to solve the problem.
  - What is the least amount of time required to travel between two stops?
  - What is the minimum distance that can be traveled to reach a stop from another stop?
  - What is the least fare required to travel between two stops if fares between stops are added to give the total fare?

In Exercises 2–4 find the length of a shortest path between  $a$  and  $z$  in the given weighted graph.



- Find a shortest path between  $a$  and  $z$  in each of the weighted graphs in Exercises 2–4.
- Find the length of a shortest path between these pairs of vertices in the weighted graph in Exercise 3.
  - $a$  and  $d$
  - $a$  and  $f$
  - $c$  and  $f$
  - $b$  and  $z$
- Find shortest paths in the weighted graph in Exercise 3 between the pairs of vertices in Exercise 6.
- Find a shortest path (in mileage) between each of the following pairs of cities in the airline system shown in Figure 1.
  - New York and Los Angeles
  - Boston and San Francisco
  - Miami and Denver
  - Miami and Los Angeles

9. Find a combination of flights with the least total air time between the pairs of cities in Exercise 8, using the flight times shown in Figure 1.
10. Find a least expensive combination of flights connecting the pairs of cities in Exercise 8, using the fares shown in Figure 1.
11. Find a shortest route (in distance) between computer centers in each of these pairs of cities in the communications network shown in Figure 2.
  - a) Boston and Los Angeles
  - b) New York and San Francisco
  - c) Dallas and San Francisco
  - d) Denver and New York
12. Find a route with the shortest response time between the pairs of computer centers in Exercise 11 using the response times given in Figure 2.
13. Find a least expensive route, in monthly lease charges, between the pairs of computer centers in Exercise 11 using the lease charges given in Figure 2.
14. Explain how to find a path with the least number of edges between two vertices in an undirected graph by considering it as a shortest path problem in a weighted graph.
15. Extend Dijkstra's algorithm for finding the length of a shortest path between two vertices in a weighted simple connected graph so that the length of a shortest path between the vertex  $a$  and every other vertex of the graph is found.
16. Extend Dijkstra's algorithm for finding the length of a shortest path between two vertices in a weighted simple connected graph so that a shortest path between these vertices is constructed.
17. The weighted graphs in the figures here show some major roads in New Jersey. Part (a) shows the distances between cities on these roads; part (b) shows the tolls.

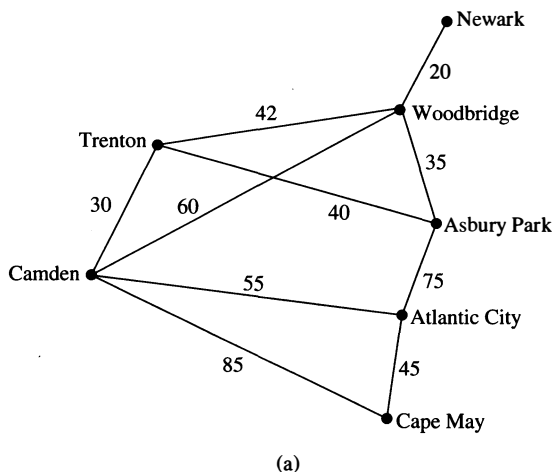


- a) Find a shortest route in distance between Newark and Camden, and between Newark and Cape May, using these roads.
  - b) Find a least expensive route in terms of total tolls using the roads in the graph between the pairs of cities in part (a) of this exercise.
18. Is a shortest path between two vertices in a weighted graph unique if the weights of edges are distinct?
  19. What are some applications where it is necessary to find the length of a longest simple path between two vertices in a weighted graph?
  20. What is the length of a longest simple path in the weighted graph in Figure 4 between  $a$  and  $z$ ? Between  $c$  and  $z$ ?



**Floyd's algorithm**, displayed as Algorithm 2, can be used to find the length of a shortest path between all pairs of vertices in a weighted connected simple graph. However, this algorithm cannot be used to construct shortest paths. (We assign an infinite weight to any pair of vertices not connected by an edge in the graph.)

21. Use Floyd's algorithm to find the distance between all pairs of vertices in the weighted graph in Figure 4(a).
- \*22. Prove that Floyd's algorithm determines the shortest distance between all pairs of vertices in a weighted simple graph.
- \*23. Give a big- $O$  estimate of the number of operations (comparisons and additions) used by Floyd's algorithm to determine the shortest distance between every pair of vertices in a weighted simple graph with  $n$  vertices.
- \*24. Show that Dijkstra's algorithm may not work if edges can have negative weights.



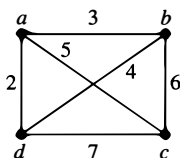
**ALGORITHM 2 Floyd's Algorithm.**

```

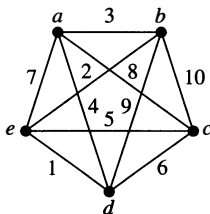
procedure Floyd( $G$ : weighted simple graph)
 $\{G$  has vertices  $v_1, v_2, \dots, v_n$  and weights  $w(v_i, v_j)$ 
  with  $w(v_i, v_j) = \infty$  if  $(v_i, v_j)$  is not an edge}
  for  $i := 1$  to  $n$ 
    for  $j := 1$  to  $n$ 
       $d(v_i, v_j) := w(v_i, v_j)$ 
  for  $i := 1$  to  $n$ 
    for  $j := 1$  to  $n$ 
      for  $k := 1$  to  $n$ 
        if  $d(v_j, v_i) + d(v_i, v_k) < d(v_j, v_k)$ 
          then  $d(v_j, v_k) :=$ 
             $d(v_j, v_i) + d(v_i, v_k)$ 
 $\{d(v_i, v_j)$  is the length of a shortest path between  $v_i$ 
and  $v_j\}$ 

```

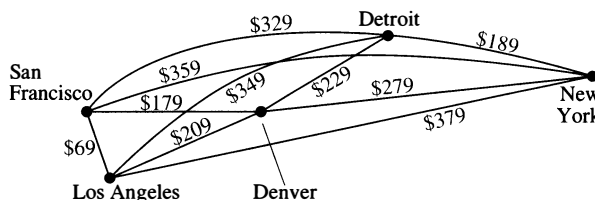
25. Solve the traveling salesman problem for this graph by finding the total weight of all Hamilton circuits and determining a circuit with minimum total weight.



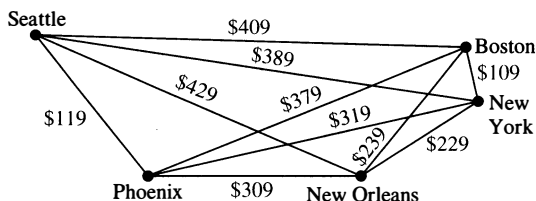
26. Solve the traveling salesman problem for this graph by finding the total weight of all Hamilton circuits and determining a circuit with minimum total weight.



27. Find a route with the least total airfare that visits each of the cities in this graph, where the weight on an edge is the least price available for a flight between the two cities.



28. Find a route with the least total airfare that visits each of the cities in this graph, where the weight on an edge is the least price available for a flight between the two cities.



29. Construct a weighted undirected graph such that the total weight of a circuit that visits every vertex at least once is minimized for a circuit that visits some vertices more than once. [Hint: There are examples with three vertices.]
30. Show that the problem of finding a circuit of minimum total weight that visits every vertex of a weighted graph at least once can be reduced to the problem of finding a circuit of minimum total weight that visits each vertex of a weighted graph exactly once. Do so by constructing a new weighted graph with the same vertices and edges as the original graph but whose weight of the edge connecting the vertices  $u$  and  $v$  is equal to the minimum total weight of a path from  $u$  to  $v$  in the original graph.

## 9.7 Planar Graphs

### Introduction



Consider the problem of joining three houses to each of three separate utilities, as shown in Figure 1. Is it possible to join these houses and utilities so that none of the connections cross? This problem can be modeled using the complete bipartite graph  $K_{3,3}$ . The original question can be rephrased as: Can  $K_{3,3}$  be drawn in the plane so that no two of its edges cross?

In this section we will study the question of whether a graph can be drawn in the plane without edges crossing. In particular, we will answer the houses-and-utilities problem.

There are always many ways to represent a graph. When is it possible to find at least one way to represent this graph in a plane without any edges crossing?

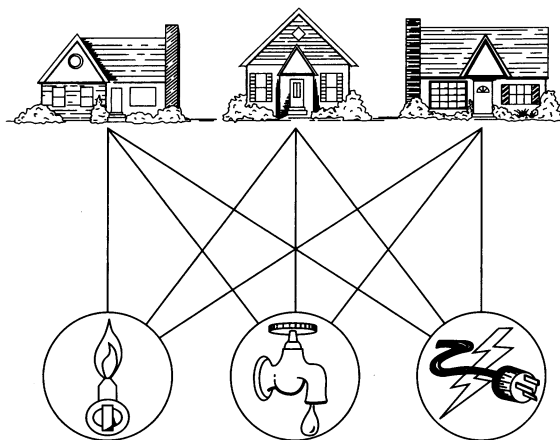


FIGURE 1 Three Houses and Three Utilities.

**DEFINITION 1** A graph is called *planar* if it can be drawn in the plane without any edges crossing (where a crossing of edges is the intersection of the lines or arcs representing them at a point other than their common endpoint). Such a drawing is called a *planar representation* of the graph.

A graph may be planar even if it is usually drawn with crossings, because it may be possible to draw it in a different way without crossings.

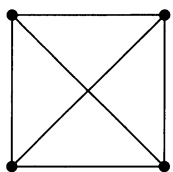
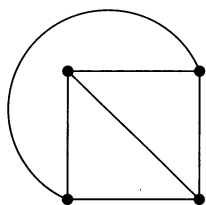
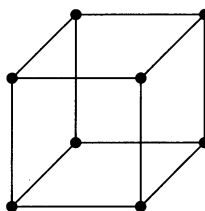
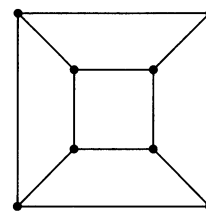
**EXAMPLE 1** Is  $K_4$  (shown in Figure 2 with two edges crossing) planar?

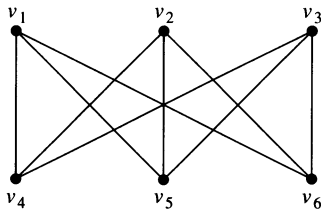
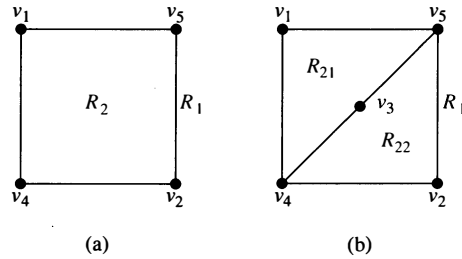
*Solution:*  $K_4$  is planar because it can be drawn without crossings, as shown in Figure 3. ◀

**EXAMPLE 2** Is  $Q_3$ , shown in Figure 4, planar?

*Solution:*  $Q_3$  is planar, because it can be drawn without any edges crossing, as shown in Figure 5. ◀

We can show that a graph is planar by displaying a planar representation. It is harder to show that a graph is nonplanar. We will give an example to show how this can be done in an ad hoc fashion. Later we will develop some general results that can be used to do this.

FIGURE 2 The Graph  $K_4$ .FIGURE 3  $K_4$  Drawn with No Crossings.FIGURE 4 The Graph  $Q_3$ .FIGURE 5 A Planar Representation of  $Q_3$ .

FIGURE 6 The Graph  $K_{3,3}$ .FIGURE 7 Showing that  $K_{3,3}$  Is Nonplanar.

**EXAMPLE 3** Is  $K_{3,3}$ , shown in Figure 6, planar?

*Solution:* Any attempt to draw  $K_{3,3}$  in the plane with no edges crossing is doomed. We now show why. In any planar representation of  $K_{3,3}$ , the vertices  $v_1$  and  $v_2$  must be connected to both  $v_4$  and  $v_5$ . These four edges form a closed curve that splits the plane into two regions,  $R_1$  and  $R_2$ , as shown in Figure 7(a). The vertex  $v_3$  is in either  $R_1$  or  $R_2$ . When  $v_3$  is in  $R_2$ , the inside of the closed curve, the edges between  $v_3$  and  $v_4$  and between  $v_3$  and  $v_5$  separate  $R_2$  into two subregions,  $R_{21}$  and  $R_{22}$ , as shown in Figure 7(b).

Next, note that there is no way to place the final vertex  $v_6$  without forcing a crossing. For if  $v_6$  is in  $R_1$ , then the edge between  $v_6$  and  $v_3$  cannot be drawn without a crossing. If  $v_6$  is in  $R_{21}$ , then the edge between  $v_2$  and  $v_6$  cannot be drawn without a crossing. If  $v_6$  is in  $R_{22}$ , then the edge between  $v_1$  and  $v_6$  cannot be drawn without a crossing.

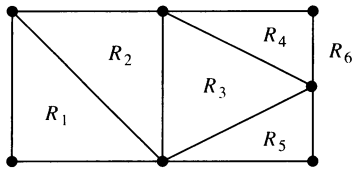
A similar argument can be used when  $v_3$  is in  $R_1$ . The completion of this argument is left for the reader (see Exercise 10 at the end of this section). It follows that  $K_{3,3}$  is not planar. ◀

Example 3 solves the utilities-and-houses problem that was described at the beginning of this section. The three houses and three utilities cannot be connected in the plane without a crossing. A similar argument can be used to show that  $K_5$  is nonplanar. (See Exercise 11 at the end of this section.)

Planarity of graphs plays an important role in the design of electronic circuits. We can model a circuit with a graph by representing components of the circuit by vertices and connections between them by edges. We can print a circuit on a single board with no connections crossing if the graph representing the circuit is planar. When this graph is not planar, we must turn to more expensive options. For example, we can partition the vertices in the graph representing the circuit into planar subgraphs. We then construct the circuit using multiple layers. (See the preamble to Exercise 30 to learn about the thickness of a graph.) We can construct the circuit using insulated wires whenever connections cross. In this case, drawing the graph with the fewest possible crossings is important. (See the preamble to Exercise 26 to learn about the crossing number of a graph.)

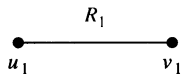
## Euler's Formula

A planar representation of a graph splits the plane into **regions**, including an unbounded region. For instance, the planar representation of the graph shown in Figure 8 splits the plane into six regions. These are labeled in the figure. Euler showed that all planar representations of a graph split the plane into the same number of regions. He accomplished this by finding a relationship among the number of regions, the number of vertices, and the number of edges of a planar graph.



**FIGURE 8** The Regions of the Planar Representation of a Graph.

**THEOREM 1 EULER'S FORMULA** Let  $G$  be a connected planar simple graph with  $e$  edges and  $v$  vertices. Let  $r$  be the number of regions in a planar representation of  $G$ . Then  $r = e - v + 2$ .



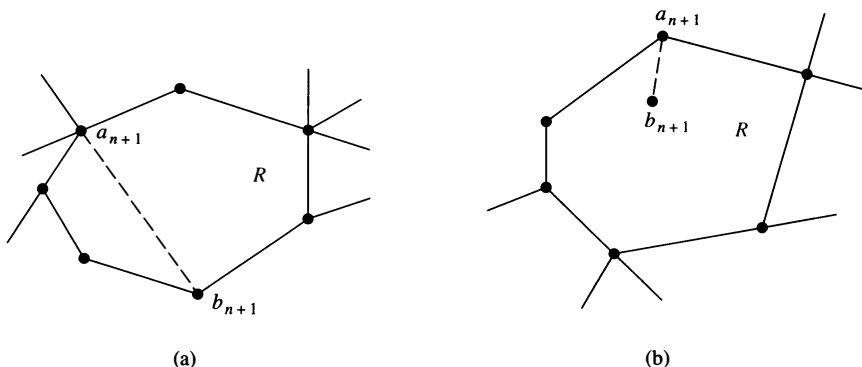
**FIGURE 9** The Basis Case of the Proof of Euler's Formula.

**Proof:** First, we specify a planar representation of  $G$ . We will prove the theorem by constructing a sequence of subgraphs  $G_1, G_2, \dots, G_e = G$ , successively adding an edge at each stage. This is done using the following inductive definition. Arbitrarily pick one edge of  $G$  to obtain  $G_1$ . Obtain  $G_n$  from  $G_{n-1}$  by arbitrarily adding an edge that is incident with a vertex already in  $G_{n-1}$ , adding the other vertex incident with this edge if it is not already in  $G_{n-1}$ . This construction is possible because  $G$  is connected.  $G$  is obtained after  $e$  edges are added. Let  $r_n, e_n$ , and  $v_n$  represent the number of regions, edges, and vertices of the planar representation of  $G_n$  induced by the planar representation of  $G$ , respectively.

The proof will now proceed by induction. The relationship  $r_1 = e_1 - v_1 + 2$  is true for  $G_1$ , because  $e_1 = 1, v_1 = 2$ , and  $r_1 = 1$ . This is shown in Figure 9.

Now assume that  $r_n = e_n - v_n + 2$ . Let  $\{a_{n+1}, b_{n+1}\}$  be the edge that is added to  $G_n$  to obtain  $G_{n+1}$ . There are two possibilities to consider. In the first case, both  $a_{n+1}$  and  $b_{n+1}$  are already in  $G_n$ . These two vertices must be on the boundary of a common region  $R$ , or else it would be impossible to add the edge  $\{a_{n+1}, b_{n+1}\}$  to  $G_n$  without two edges crossing (and  $G_{n+1}$  is planar). The addition of this new edge splits  $R$  into two regions. Consequently, in this case,  $r_{n+1} = r_n + 1$ ,  $e_{n+1} = e_n + 1$ , and  $v_{n+1} = v_n$ . Thus, each side of the formula relating the number of regions, edges, and vertices increases by exactly one, so this formula is still true. In other words,  $r_{n+1} = e_{n+1} - v_{n+1} + 2$ . This case is illustrated in Figure 10(a).

In the second case, one of the two vertices of the new edge is not already in  $G_n$ . Suppose that  $a_{n+1}$  is in  $G_n$  but that  $b_{n+1}$  is not. Adding this new edge does not produce any new regions, because  $b_{n+1}$  must be in a region that has  $a_{n+1}$  on its boundary. Consequently,  $r_{n+1} = r_n$ . Moreover,  $e_{n+1} = e_n + 1$  and  $v_{n+1} = v_n + 1$ . Each side of the formula relating the number



**FIGURE 10** Adding an Edge to  $G_n$  to Produce  $G_{n+1}$ .



of regions, edges, and vertices remains the same, so the formula is still true. In other words,  $r_{n+1} = e_{n+1} - v_{n+1} + 2$ . This case is illustrated in Figure 10(b).

We have completed the induction argument. Hence,  $r_n = e_n - v_n + 2$  for all  $n$ . Because the original graph is the graph  $G_e$ , obtained after  $e$  edges have been added, the theorem is true. ◀

Euler's formula is illustrated in Example 4.

**EXAMPLE 4** Suppose that a connected planar simple graph has 20 vertices, each of degree 3. Into how many regions does a representation of this planar graph split the plane?

*Solution:* This graph has 20 vertices, each of degree 3, so  $v = 20$ . Because the sum of the degrees of the vertices,  $3v = 3 \cdot 20 = 60$ , is equal to twice the number of edges,  $2e$ , we have  $2e = 60$ , or  $e = 30$ . Consequently, from Euler's formula, the number of regions is

$$r = e - v + 2 = 30 - 20 + 2 = 12. \quad \blacktriangleleft$$

Euler's formula can be used to establish some inequalities that must be satisfied by planar graphs. One such inequality is given in Corollary 1.

**COROLLARY 1** If  $G$  is a connected planar simple graph with  $e$  edges and  $v$  vertices, where  $v \geq 3$ , then  $e \leq 3v - 6$ .

Before we prove Corollary 1 we will use it to prove the following useful result.

**COROLLARY 2** If  $G$  is a connected planar simple graph, then  $G$  has a vertex of degree not exceeding five.

*Proof:* If  $G$  has one or two vertices, the result is true. If  $G$  has at least three vertices, by Corollary 1 we know that  $e \leq 3v - 6$ , so  $2e \leq 6v - 12$ . If the degree of every vertex were at least six, then because  $2e = \sum_{v \in V} \deg(v)$  (by the Handshaking Theorem), we would have  $2e \geq 6v$ . But this contradicts the inequality  $2e \leq 6v - 12$ . It follows that there must be a vertex with degree no greater than five. ◀

The proof of Corollary 1 is based on the concept of the **degree** of a region, which is defined to be the number of edges on the boundary of this region. When an edge occurs twice on the boundary (so that it is traced out twice when the boundary is traced out), it contributes two to the degree. The degrees of the regions of the graph shown in Figure 11 are displayed in the figure.

The proof of Corollary 1 can now be given.

*Proof:* A connected planar simple graph drawn in the plane divides the plane into regions, say  $r$  of them. The degree of each region is at least three. (Because the graphs discussed here are simple graphs, no multiple edges that could produce regions of degree two, or loops that could produce regions of degree one, are permitted.) In particular, note that the degree of the unbounded region is at least three because there are at least three vertices in the graph.

Note that the sum of the degrees of the regions is exactly twice the number of edges in the graph, because each edge occurs on the boundary of a region exactly twice (either in two

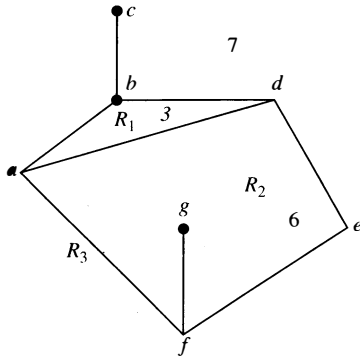


FIGURE 11 The Degrees of Regions.

different regions, or twice in the same region). Because each region has degree greater than or equal to three, it follows that

$$2e = \sum_{\text{all regions } R} \deg(R) \geq 3r.$$

Hence,

$$(2/3)e \geq r.$$

Using  $r = e - v + 2$  (Euler's formula), we obtain

$$e - v + 2 \leq (2/3)e.$$

It follows that  $e/3 \leq v - 2$ . This shows that  $e \leq 3v - 6$ . ◀

This corollary can be used to demonstrate that  $K_5$  is nonplanar.

**EXAMPLE 5** Show that  $K_5$  is nonplanar using Corollary 1.

*Solution:* The graph  $K_5$  has five vertices and 10 edges. However, the inequality  $e \leq 3v - 6$  is not satisfied for this graph because  $e = 10$  and  $3v - 6 = 9$ . Therefore,  $K_5$  is not planar. ◀

It was previously shown that  $K_{3,3}$  is not planar. Note, however, that this graph has six vertices and nine edges. This means that the inequality  $e = 9 \leq 12 = 3 \cdot 6 - 6$  is satisfied. Consequently, the fact that the inequality  $e \leq 3v - 6$  is satisfied does *not* imply that a graph is planar. However, the following corollary of Theorem 1 can be used to show that  $K_{3,3}$  is nonplanar.

**COROLLARY 3** If a connected planar simple graph has  $e$  edges and  $v$  vertices with  $v \geq 3$  and no circuits of length three, then  $e \leq 2v - 4$ .

The proof of Corollary 3 is similar to that of Corollary 1, except that in this case the fact that there are no circuits of length three implies that the degree of a region must be at least four. The details of this proof are left for the reader (see Exercise 15 at the end of this section).

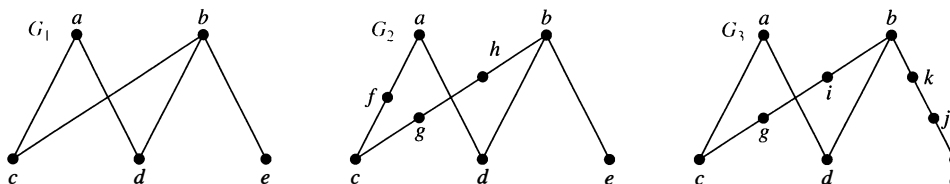


FIGURE 12 Homeomorphic Graphs.

**EXAMPLE 6** Use Corollary 3 to show that  $K_{3,3}$  is nonplanar.

*Solution:* Because  $K_{3,3}$  has no circuits of length three (this is easy to see because it is bipartite), Corollary 3 can be used.  $K_{3,3}$  has six vertices and nine edges. Because  $e = 9$  and  $2v - 4 = 8$ , Corollary 3 shows that  $K_{3,3}$  is nonplanar.  $\blacktriangleleft$

## Kuratowski's Theorem

We have seen that  $K_{3,3}$  and  $K_5$  are not planar. Clearly, a graph is not planar if it contains either of these two graphs as a subgraph. Surprisingly, all nonplanar graphs must contain a subgraph that can be obtained from  $K_{3,3}$  or  $K_5$  using certain permitted operations.

If a graph is planar, so will be any graph obtained by removing an edge  $\{u, v\}$  and adding a new vertex  $w$  together with edges  $\{u, w\}$  and  $\{w, v\}$ . Such an operation is called an **elementary subdivision**. The graphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  are called **homeomorphic** if they can be obtained from the same graph by a sequence of elementary subdivisions.

**EXAMPLE 7** Show that the graphs  $G_1$ ,  $G_2$ , and  $G_3$  displayed in Figure 12 are all homeomorphic.

*Solution:* These three graphs are homeomorphic because all three can be obtained from  $G_1$  by elementary subdivisions.  $G_1$  can be obtained from itself by an empty sequence of elementary subdivisions. To obtain  $G_2$  from  $G_1$  we can use this sequence of elementary subdivisions: (i) remove the edge  $\{a, c\}$ , add the vertex  $f$ , and add the edges  $\{a, f\}$  and  $\{f, c\}$ ; (ii) remove the edge  $\{b, c\}$ , add the vertex  $g$ , and add the edges  $\{b, g\}$  and  $\{g, c\}$ ; and (iii) remove the edge  $\{b, g\}$ , add the vertex  $h$ , and add the edges  $\{g, h\}$  and  $\{b, h\}$ . We leave it to the reader to determine the sequence of elementary subdivisions needed to obtain  $G_3$  from  $G_1$ .  $\blacktriangleleft$

The Polish mathematician Kazimierz Kuratowski established Theorem 2 in 1930, which characterizes planar graphs using the concept of graph homeomorphism.



**KAZIMIERZ KURATOWSKI (1896–1980)** Kazimierz Kuratowski, the son of a famous Warsaw lawyer, attended secondary school in Warsaw. He studied in Glasgow, Scotland, from 1913 to 1914 but could not return there after the outbreak of World War I. In 1915 he entered Warsaw University, where he was active in the Polish patriotic student movement. He published his first paper in 1919 and received his Ph.D. in 1921. He was an active member of the group known as the Warsaw School of Mathematics, working in the areas of the foundations of set theory and topology. He was appointed associate professor at the Lwów Polytechnical University, where he stayed for seven years, collaborating with the important Polish mathematicians Banach and Ulam. In 1930, while at Lwów, Kuratowski completed his work characterizing planar graphs.

In 1934 he returned to Warsaw University as a full professor. Until the start of World War II, he was active in research and teaching. During the war, because of the persecution of educated Poles, Kuratowski went into hiding under an assumed name and taught at the clandestine Warsaw University. After the war he helped revive Polish mathematics, serving as director of the Polish National Mathematics Institute. He wrote over 180 papers and three widely used textbooks.

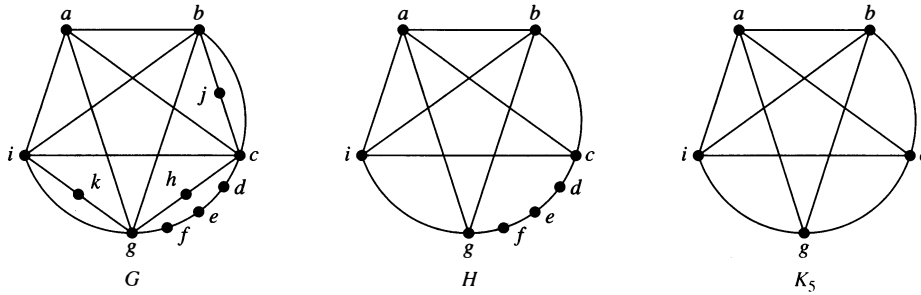


FIGURE 13 The Undirected Graph  $G$ , a Subgraph  $H$  Homeomorphic to  $K_5$ , and  $K_5$ .

**THEOREM 2** A graph is nonplanar if and only if it contains a subgraph homeomorphic to  $K_{3,3}$  or  $K_5$ .

It is clear that a graph containing a subgraph homeomorphic to  $K_{3,3}$  or  $K_5$  is nonplanar. However, the proof of the converse, namely that every nonplanar graph contains a subgraph homeomorphic to  $K_{3,3}$  or  $K_5$ , is complicated and will not be given here. Examples 8 and 9 illustrate how Kuratowski's Theorem is used.

**EXAMPLE 8** Determine whether the graph  $G$  shown in Figure 13 is planar.



**Solution:**  $G$  has a subgraph  $H$  homeomorphic to  $K_5$ .  $H$  is obtained by deleting  $h$ ,  $j$ , and  $k$  and all edges incident with these vertices.  $H$  is homeomorphic to  $K_5$  because it can be obtained from  $K_5$  (with vertices  $a$ ,  $b$ ,  $c$ ,  $g$ , and  $i$ ) by a sequence of elementary subdivisions, adding the vertices  $d$ ,  $e$ , and  $f$ . (The reader should construct such a sequence of elementary subdivisions.) Hence,  $G$  is nonplanar. ◀

**EXAMPLE 9** Is the Petersen graph, shown in Figure 14(a), planar? (The Danish mathematician Julius Petersen studied this graph in 1891; it is often used to illustrate various theoretical properties of graphs.)

**Solution:** The subgraph  $H$  of the Petersen graph obtained by deleting  $b$  and the three edges that have  $b$  as an endpoint, shown in Figure 14(b), is homeomorphic to  $K_{3,3}$ , with vertex sets

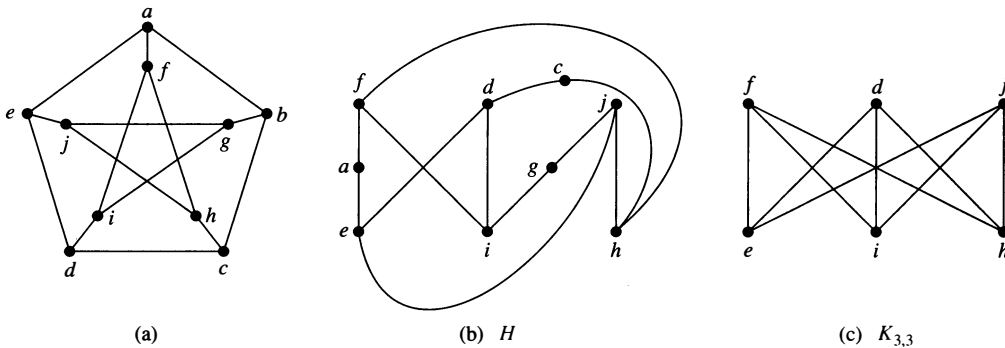


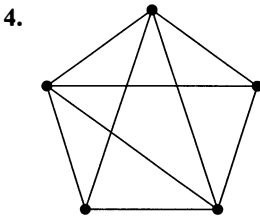
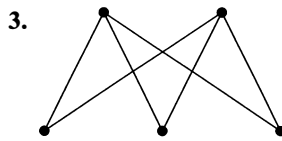
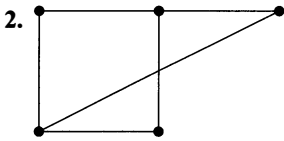
FIGURE 14 (a) The Petersen Graph, (b) a Subgraph  $H$  Homeomorphic to  $K_{3,3}$ , and (c)  $K_{3,3}$ .

$\{f, d, j\}$  and  $\{e, i, h\}$ , because it can be obtained by a sequence of elementary subdivisions, deleting  $\{d, h\}$  and adding  $\{c, h\}$  and  $\{c, d\}$ , deleting  $\{e, f\}$  and adding  $\{a, e\}$  and  $\{a, f\}$ , and deleting  $\{i, j\}$  and adding  $\{g, i\}$  and  $\{g, j\}$ . Hence, the Petersen graph is not planar. ◀

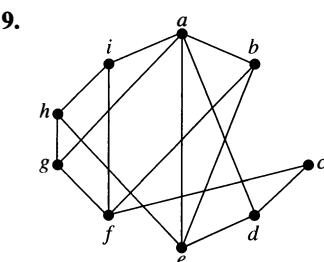
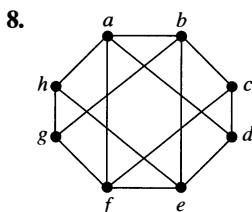
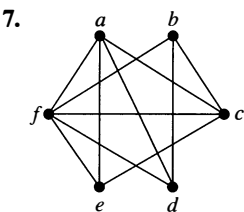
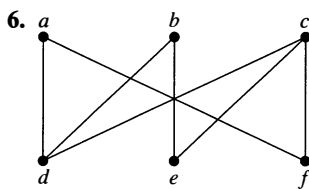
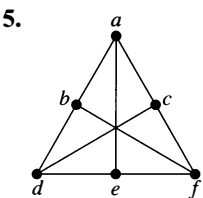
## Exercises

1. Can five houses be connected to two utilities without connections crossing?

In Exercises 2–4 draw the given planar graph without any crossings.



In Exercises 5–9 determine whether the given graph is planar. If so, draw it so that no edges cross.



10. Complete the argument in Example 3.

11. Show that  $K_5$  is nonplanar using an argument similar to that given in Example 3.

12. Suppose that a connected planar graph has eight vertices, each of degree three. Into how many regions is the plane divided by a planar representation of this graph?

13. Suppose that a connected planar graph has six vertices, each of degree four. Into how many regions is the plane divided by a planar representation of this graph?

14. Suppose that a connected planar graph has 30 edges. If a planar representation of this graph divides the plane into 20 regions, how many vertices does this graph have?

15. Prove Corollary 3.

16. Suppose that a connected bipartite planar simple graph has  $e$  edges and  $v$  vertices. Show that  $e \leq 2v - 4$  if  $v \geq 3$ .

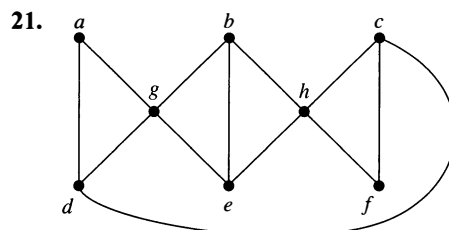
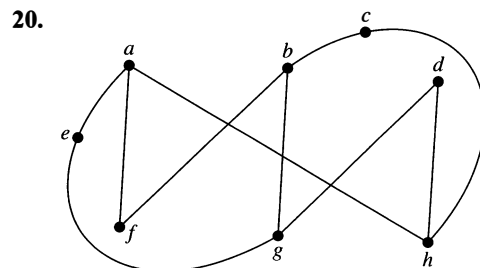
- \*17. Suppose that a connected planar simple graph with  $e$  edges and  $v$  vertices contains no simple circuits of length 4 or less. Show that  $e \leq (5/3)v - (10/3)$  if  $v \geq 4$ .

18. Suppose that a planar graph has  $k$  connected components,  $e$  edges, and  $v$  vertices. Also suppose that the plane is divided into  $r$  regions by a planar representation of the graph. Find a formula for  $r$  in terms of  $e$ ,  $v$ , and  $k$ .

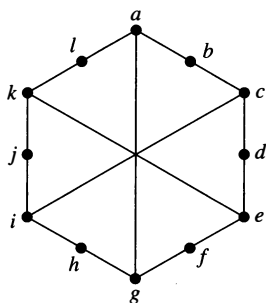
19. Which of these nonplanar graphs have the property that the removal of any vertex and all edges incident with that vertex produces a planar graph?

a)  $K_5$    b)  $K_6$    c)  $K_{3,3}$    d)  $K_{3,4}$

In Exercises 20–22 determine whether the given graph is homeomorphic to  $K_{3,3}$ .

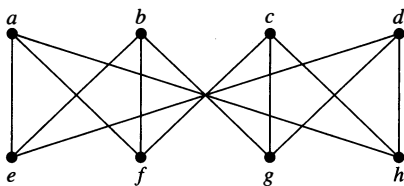


22.

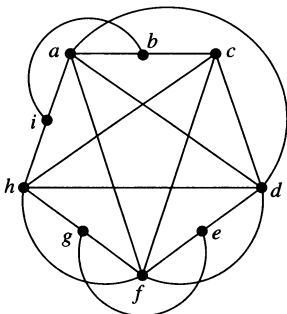


In Exercises 23–25 use Kuratowski's Theorem to determine whether the given graph is planar.

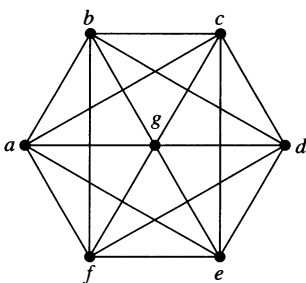
23.



24.



25.



The **crossing number** of a simple graph is the minimum number of crossings that can occur when this graph is drawn in the plane where no three arcs representing edges are permitted to cross at the same point.

26. Show that  $K_{3,3}$  has 1 as its crossing number.

**\*\*27.** Find the crossing numbers of each of these nonplanar graphs.

- |              |              |              |
|--------------|--------------|--------------|
| a) $K_5$     | b) $K_6$     | c) $K_7$     |
| d) $K_{3,4}$ | e) $K_{4,4}$ | f) $K_{5,5}$ |

**\*28.** Find the crossing number of the Petersen graph.

**\*\*29.** Show that if  $m$  and  $n$  are even positive integers, the crossing number of  $K_{m,n}$  is less than or equal to  $mn(m-2)(n-2)/16$ . [Hint: Place  $m$  vertices along the  $x$ -axis so that they are equally spaced and symmetric about the origin and place  $n$  vertices along the  $y$ -axis so that they are equally spaced and symmetric about the origin. Now connect each of the  $m$  vertices on the  $x$ -axis to each of the vertices on the  $y$ -axis and count the crossings.]

The **thickness** of a simple graph  $G$  is the smallest number of planar subgraphs of  $G$  that have  $G$  as their union.

30. Show that  $K_{3,3}$  has 2 as its thickness.

**\*31.** Find the thickness of the graphs in Exercise 27.

32. Show that if  $G$  is a connected simple graph with  $v$  vertices and  $e$  edges, then the thickness of  $G$  is at least  $\lceil e/(3v-6) \rceil$ .

**\*33.** Use Exercise 32 to show that the thickness of  $K_n$  is at least  $\lfloor (n+7)/6 \rfloor$  whenever  $n$  is a positive integer.

34. Show that if  $G$  is a connected simple graph with  $v$  vertices and  $e$  edges and no circuits of length three, then the thickness of  $G$  is at least  $\lceil e/(2v-4) \rceil$ .

35. Use Exercise 34 to show that the thickness of  $K_{m,n}$  is at least  $\lceil mn/(2m+2n-4) \rceil$  whenever  $m$  and  $n$  are positive integers.

**\*36.** Draw  $K_5$  on the surface of a torus (a doughnut-shaped solid) so that no edges cross.

**\*37.** Draw  $K_{3,3}$  on the surface of a torus so that no edges cross.

## 9.8 Graph Coloring

### Introduction

Links



Problems related to the coloring of maps of regions, such as maps of parts of the world, have generated many results in graph theory. When a map\* is colored, two regions with a common border are customarily assigned different colors. One way to ensure that two adjacent regions never have the same color is to use a different color for each region. However, this is inefficient,

\*We will assume that all regions in a map are connected. This eliminates any problems presented by such geographical entities

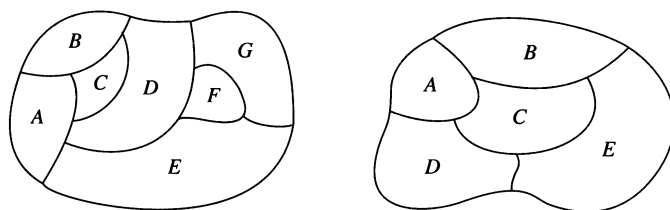


FIGURE 1 Two Maps.

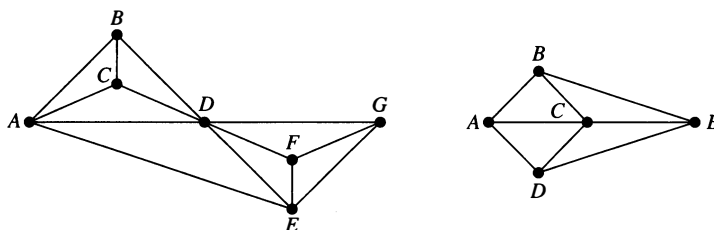


FIGURE 2 Dual Graphs of the Maps in Figure 1.

and on maps with many regions it would be hard to distinguish similar colors. Instead, a small number of colors should be used whenever possible. Consider the problem of determining the least number of colors that can be used to color a map so that adjacent regions never have the same color. For instance, for the map shown on the left in Figure 1, four colors suffice, but three colors are not enough. (The reader should check this.) In the map on the right in Figure 1, three colors are sufficient (but two are not).

Each map in the plane can be represented by a graph. To set up this correspondence, each region of the map is represented by a vertex. Edges connect two vertices if the regions represented by these vertices have a common border. Two regions that touch at only one point are not considered adjacent. The resulting graph is called the **dual graph** of the map. By the way in which dual graphs of maps are constructed, it is clear that any map in the plane has a planar dual graph. Figure 2 displays the dual graphs that correspond to the maps shown in Figure 1.

The problem of coloring the regions of a map is equivalent to the problem of coloring the vertices of the dual graph so that no two adjacent vertices in this graph have the same color. We now define a graph coloring.

**DEFINITION 1** A *coloring* of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.

A graph can be colored by assigning a different color to each of its vertices. However, for most graphs a coloring can be found that uses fewer colors than the number of vertices in the graph. What is the least number of colors necessary?

**DEFINITION 2** The *chromatic number* of a graph is the least number of colors needed for a coloring of this graph. The chromatic number of a graph  $G$  is denoted by  $\chi(G)$ . (Here  $\chi$  is the Greek letter *chi*.)

Note that asking for the chromatic number of a planar graph is the same as asking for the minimum number of colors required to color a planar map so that no two adjacent regions are assigned the same color. This question has been studied for more than 100 years. The answer is provided by one of the most famous theorems in mathematics.

**THEOREM 1 THE FOUR COLOR THEOREM** The chromatic number of a planar graph is no greater than four.



The Four Color Theorem was originally posed as a conjecture in the 1850s. It was finally proved by the American mathematicians Kenneth Appel and Wolfgang Haken in 1976. Prior to 1976, many incorrect proofs were published, often with hard-to-find errors. In addition, many futile attempts were made to construct counterexamples by drawing maps that require more than four colors. (Proving the Five Color Theorem is not that difficult; see Exercise 36.)

Perhaps the most notorious fallacious proof in all of mathematics is the incorrect proof of the Four Color Theorem published in 1879 by a London barrister and amateur mathematician, Alfred Kempe. Mathematicians accepted his proof as correct until 1890, when Percy Heawood found an error that made Kempe's argument incomplete. However, Kempe's line of reasoning turned out to be the basis of the successful proof given by Appel and Haken. Their proof relies on a careful case-by-case analysis carried out by computer. They showed that if the Four Color Theorem were false, there would have to be a counterexample of one of approximately 2000 different types, and they then showed that none of these types exists. They used over 1000 hours of computer time in their proof. This proof generated a large amount of controversy, because computers played such an important role in it. For example, could there be an error in a computer program that led to incorrect results? Was their argument really a proof if it depended on what could be unreliable computer output?

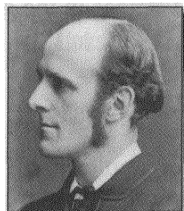
Note that the Four Color Theorem applies only to planar graphs. Nonplanar graphs can have arbitrarily large chromatic numbers, as will be shown in Example 2.

Two things are required to show that the chromatic number of a graph is  $k$ . First, we must show that the graph can be colored with  $k$  colors. This can be done by constructing such a coloring. Second, we must show that the graph cannot be colored using fewer than  $k$  colors. Examples 1–4 illustrate how chromatic numbers can be found.

**EXAMPLE 1** What are the chromatic numbers of the graphs  $G$  and  $H$  shown in Figure 3?

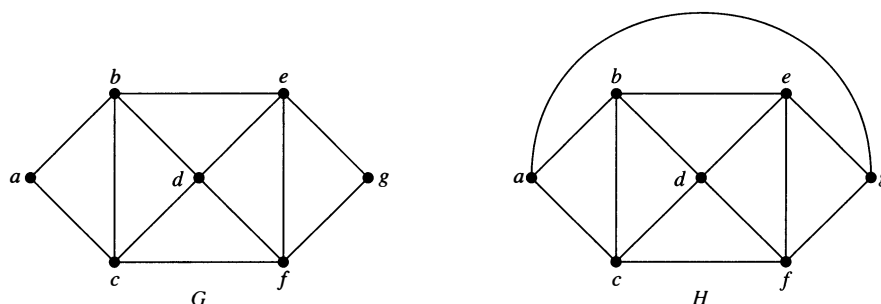
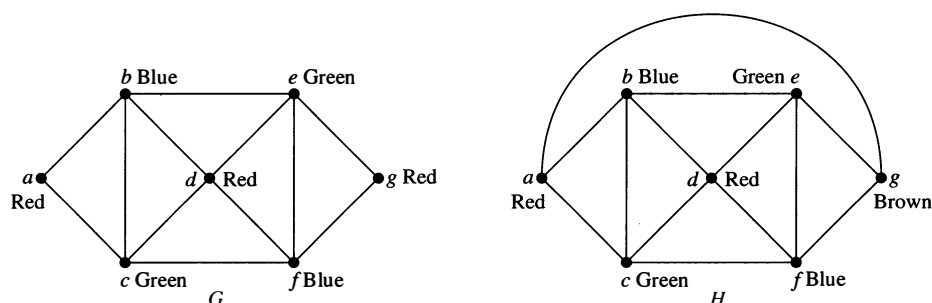


*Solution:* The chromatic number of  $G$  is at least three, because the vertices  $a$ ,  $b$ , and  $c$  must be assigned different colors. To see if  $G$  can be colored with three colors, assign red to  $a$ , blue to  $b$ , and green to  $c$ . Then,  $d$  can (and must) be colored red because it is adjacent to  $b$  and  $c$ . Furthermore,  $e$  can (and must) be colored green because it is adjacent only to vertices colored red and blue, and  $f$  can (and must) be colored blue because it is adjacent only to vertices colored red and green. Finally,  $g$  can (and must) be colored red because it is adjacent only to vertices



**ALFRED BRAY KEMPE (1849–1922)** Kempe was a barrister and a leading authority on ecclesiastical law. However, having studied mathematics at Cambridge University, he retained his interest in it, and later in life he devoted considerable time to mathematical research. Kempe made contributions to kinematics, the branch of mathematics dealing with motion, and to mathematical logic. However, Kempe is best remembered for his fallacious proof of the Four Color Theorem.



FIGURE 3 The Simple Graphs  $G$  and  $H$ .FIGURE 4 Colorings of the Graphs  $G$  and  $H$ .

colored blue and green. This produces a coloring of  $G$  using exactly three colors. Figure 4 displays such a coloring.

The graph  $H$  is made up of the graph  $G$  with an edge connecting  $a$  and  $g$ . Any attempt to color  $H$  using three colors must follow the same reasoning as that used to color  $G$ , except at the last stage, when all vertices other than  $g$  have been colored. Then, because  $g$  is adjacent (in  $H$ ) to vertices colored red, blue, and green, a fourth color, say brown, needs to be used. Hence,  $H$  has a chromatic number equal to 4. A coloring of  $H$  is shown in Figure 4. ◀

## EXAMPLE 2 What is the chromatic number of $K_n$ ?

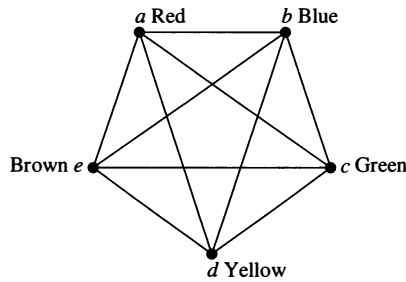
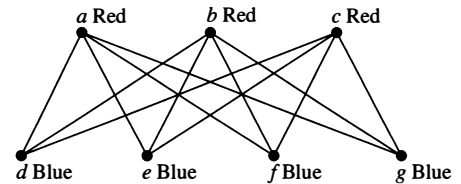
**Solution:** A coloring of  $K_n$  can be constructed using  $n$  colors by assigning a different color to each vertex. Is there a coloring using fewer colors? The answer is no. No two vertices can be assigned the same color, because every two vertices of this graph are adjacent. Hence, the chromatic number of  $K_n = n$ . That is,  $\chi(K_n) = n$ . (Recall that  $K_n$  is not planar when  $n \geq 5$ , so

---

**HISTORICAL NOTE** In 1852, an ex-student of Augustus De Morgan, Francis Guthrie, noticed that the counties in England could be colored using four colors so that no adjacent counties were assigned the same color. On this evidence, he conjectured that the Four Color Theorem was true. Francis told his brother Frederick, at that time a student of De Morgan, about this problem. Frederick in turn asked his teacher De Morgan about his brother's conjecture. De Morgan was extremely interested in this problem and publicized it throughout the mathematical community. In fact, the first written reference to the conjecture can be found in a letter from De Morgan to Sir William Rowan Hamilton. Although De Morgan thought Hamilton would be interested in this problem, Hamilton apparently was not interested in it, because it had nothing to do with quaternions.



**HISTORICAL NOTE** Although a simpler proof of the Four Color Theorem was found by Robertson, Sanders, Seymour, and Thomas in 1996, reducing the computational part of the proof to examining 633 configurations, no proof that does not rely on extensive computation has yet been found.

FIGURE 5 A Coloring of  $K_5$ .FIGURE 6 A Coloring of  $K_{3,4}$ .

this result does not contradict the Four Color Theorem.) A coloring of  $K_5$  using five colors is shown in Figure 5. ◀

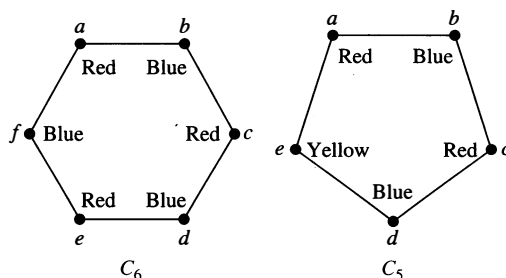
**EXAMPLE 3** What is the chromatic number of the complete bipartite graph  $K_{m,n}$ , where  $m$  and  $n$  are positive integers?

*Solution:* The number of colors needed may seem to depend on  $m$  and  $n$ . However, as Theorem 4 in Section 9.2 tells us, only two colors are needed, because  $K_{m,n}$  is a bipartite graph. Hence,  $\chi(K_{m,n}) = 2$ . This means that we can color the set of  $m$  vertices with one color and the set of  $n$  vertices with a second color. Because edges connect only a vertex from the set of  $m$  vertices and a vertex from the set of  $n$  vertices, no two adjacent vertices have the same color. A coloring of  $K_{3,4}$  with two colors is displayed in Figure 6. ◀

**EXAMPLE 4** What is the chromatic number of the graph  $C_n$ , where  $n \geq 3$ ? (Recall that  $C_n$  is the cycle with  $n$  vertices.)

*Solution:* We will first consider some individual cases. To begin, let  $n = 6$ . Pick a vertex and color it red. Proceed clockwise in the planar depiction of  $C_6$  shown in Figure 7. It is necessary to assign a second color, say blue, to the next vertex reached. Continue in the clockwise direction; the third vertex can be colored red, the fourth vertex blue, and the fifth vertex red. Finally, the sixth vertex, which is adjacent to the first, can be colored blue. Hence, the chromatic number of  $C_6$  is 2. Figure 7 displays the coloring constructed here.

Next, let  $n = 5$  and consider  $C_5$ . Pick a vertex and color it red. Proceeding clockwise, it is necessary to assign a second color, say blue, to the next vertex reached. Continuing in the clockwise direction, the third vertex can be colored red, and the fourth vertex can be colored blue.

FIGURE 7 Colorings of  $C_5$  and  $C_6$ .

The fifth vertex cannot be colored either red or blue, because it is adjacent to the fourth vertex and the first vertex. Consequently, a third color is required for this vertex. Note that we would have also needed three colors if we had colored vertices in the counterclockwise direction. Thus, the chromatic number of  $C_5$  is 3. A coloring of  $C_5$  using three colors is displayed in Figure 7.

In general, two colors are needed to color  $C_n$  when  $n$  is even. To construct such a coloring, simply pick a vertex and color it red. Proceed around the graph in a clockwise direction (using a planar representation of the graph) coloring the second vertex blue, the third vertex red, and so on. The  $n$ th vertex can be colored blue, because the two vertices adjacent to it, namely the  $(n - 1)$ st and the first vertices, are both colored red.

When  $n$  is odd and  $n > 1$ , the chromatic number of  $C_n$  is 3. To see this, pick an initial vertex. To use only two colors, it is necessary to alternate colors as the graph is traversed in a clockwise direction. However, the  $n$ th vertex reached is adjacent to two vertices of different colors, namely, the first and  $(n - 1)$ st. Hence, a third color must be used.

We have shown that  $\chi(C_n) = 2$  if  $n$  is an even positive integer with  $n \geq 4$  and  $\chi(C_n) = 3$  if  $n$  is an odd positive integer with  $n \geq 3$ . ◀



The best algorithms known for finding the chromatic number of a graph have exponential worst-case time complexity (in the number of vertices of the graph). Even the problem of finding an approximation to the chromatic number of a graph is difficult. It has been shown that if there were an algorithm with polynomial worst-case time complexity that could approximate the chromatic number of a graph up to a factor of 2 (that is, construct a bound that was no more than double the chromatic number of the graph), then an algorithm with polynomial worst-case time complexity for finding the chromatic number of the graph would also exist.

## Applications of Graph Colorings

Graph coloring has a variety of applications to problems involving scheduling and assignments. (Note that because no efficient algorithm is known for graph coloring, this does not lead to efficient algorithms for scheduling and assignments.) Examples of such applications will be given here. The first application deals with the scheduling of final exams.

**EXAMPLE 5 Scheduling Final Exams** How can the final exams at a university be scheduled so that no student has two exams at the same time?

*Solution:* This scheduling problem can be solved using a graph model, with vertices representing courses and with an edge between two vertices if there is a common student in the courses they represent. Each time slot for a final exam is represented by a different color. A scheduling of the exams corresponds to a coloring of the associated graph.

For instance, suppose there are seven finals to be scheduled. Suppose the courses are numbered 1 through 7. Suppose that the following pairs of courses have common students: 1 and 2, 1 and 3, 1 and 4, 1 and 7, 2 and 3, 2 and 4, 2 and 5, 2 and 7, 3 and 4, 3 and 6, 3 and 7, 4 and 5, 4 and 6, 5 and 6, 5 and 7, and 6 and 7. In Figure 8 the graph associated with this set of classes is shown. A scheduling consists of a coloring of this graph.

Because the chromatic number of this graph is 4 (the reader should verify this), four time slots are needed. A coloring of the graph using four colors and the associated schedule are shown in Figure 9. ◀

Now consider an application to the assignment of television channels.

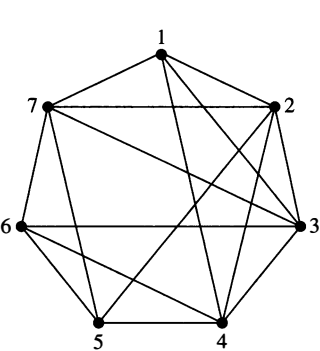
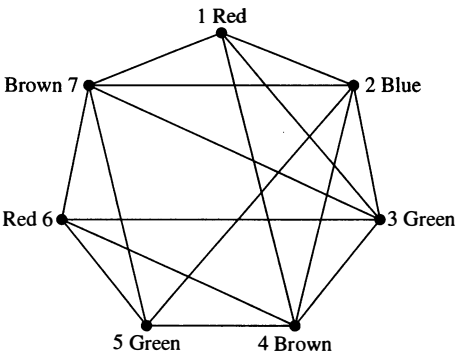


FIGURE 8 The Graph Representing the Scheduling of Final Exams.



Time Period	Courses
I	1, 6
II	2
III	3, 5
IV	4, 7

FIGURE 9 Using a Coloring to Schedule Final Exams.

**EXAMPLE 6 Frequency Assignments** Television channels 2 through 13 are assigned to stations in North America so that no two stations within 150 miles can operate on the same channel. How can the assignment of channels be modeled by graph coloring?

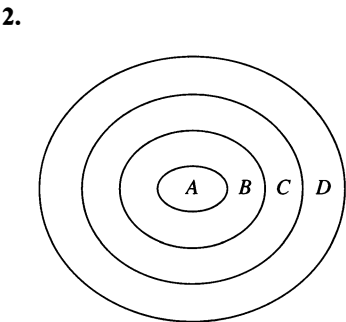
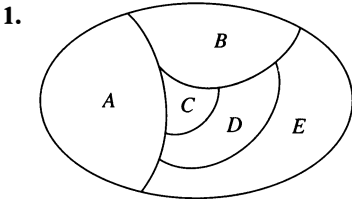
*Solution:* Construct a graph by assigning a vertex to each station. Two vertices are connected by an edge if they are located within 150 miles of each other. An assignment of channels corresponds to a coloring of the graph, where each color represents a different channel. ◀

An application of graph coloring to compilers is considered in Example 7.

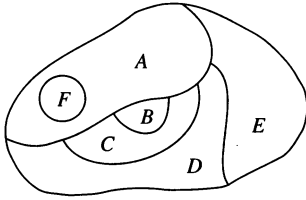
**EXAMPLE 7 Index Registers** In efficient compilers the execution of loops is speeded up when frequently used variables are stored temporarily in index registers in the central processing unit, instead of in regular memory. For a given loop, how many index registers are needed? This problem can be addressed using a graph coloring model. To set up the model, let each vertex of a graph represent a variable in the loop. There is an edge between two vertices if the variables they represent must be stored in index registers at the same time during the execution of the loop. Thus, the chromatic number of the graph gives the number of index registers needed, because different registers must be assigned to variables when the vertices representing these variables are adjacent in the graph. ◀

Exercises

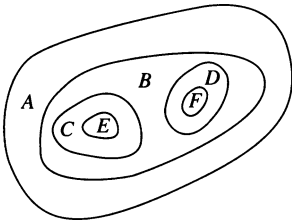
In Exercises 1–4 construct the dual graph for the map shown. Then find the number of colors needed to color the map so that no two adjacent regions have the same color.



3.

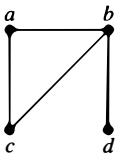


4.

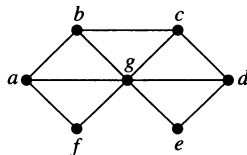


In Exercises 5–11 find the chromatic number of the given graph.

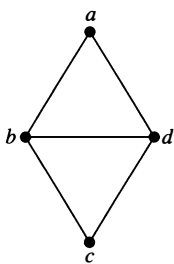
5.



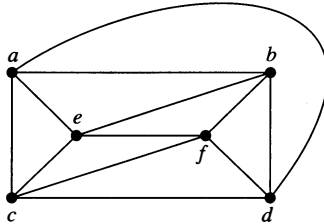
6.



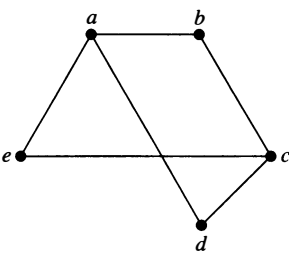
7.



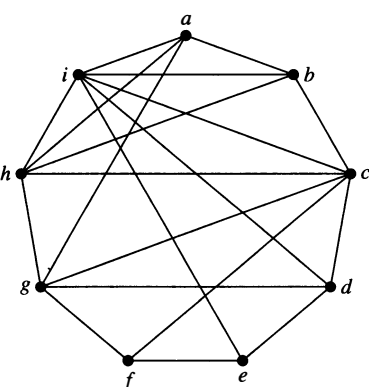
8.



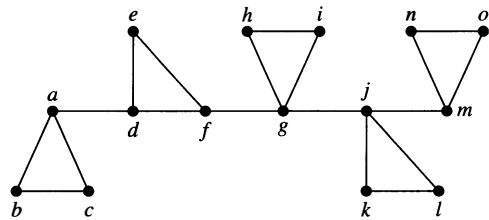
9.



10.



11.



12. For the graphs in Exercises 5–11, decide whether it is possible to decrease the chromatic number by removing a single vertex and all edges incident with it.

13. Which graphs have a chromatic number of 1?

14. What is the least number of colors needed to color a map of the United States? Do not consider adjacent states that meet only at a corner. Suppose that Michigan is one region. Consider the vertices representing Alaska and Hawaii as isolated vertices.

15. What is the chromatic number of  $W_n$ ?

16. Show that a simple graph that has a circuit with an odd number of vertices in it cannot be colored using two colors.

17. Schedule the final exams for Math 115, Math 116, Math 185, Math 195, CS 101, CS 102, CS 273, and CS 473, using the fewest number of different time slots, if there are no students taking both Math 115 and CS 473, both Math 116 and CS 473, both Math 195 and CS 101, both Math 195 and CS 102, both Math 115 and Math 116, both Math 115 and Math 185, and both Math 185 and Math 195, but there are students in every other combination of courses.

18. How many different channels are needed for six stations located at the distances shown in the table, if two stations cannot use the same channel when they are within 150 miles of each other?

	1	2	3	4	5	6
1	—	85	175	200	50	100
2	85	—	125	175	100	160
3	175	125	—	100	200	250
4	200	175	100	—	210	220
5	50	100	200	210	—	100
6	100	160	250	220	100	—

19. The mathematics department has six committees each meeting once a month. How many different meeting times must be used to ensure that no member is scheduled to attend two meetings at the same time if the committees are  $C_1 = \{\text{Arlinghaus, Brand, Zaslavsky}\}$ ,  $C_2 = \{\text{Brand, Lee, Rosen}\}$ ,  $C_3 = \{\text{Arlinghaus, Rosen, Zaslavsky}\}$ ,  $C_4 = \{\text{Lee, Rosen, Zaslavsky}\}$ ,  $C_5 = \{\text{Arlinghaus, Brand}\}$ , and  $C_6 = \{\text{Brand, Rosen, Zaslavsky}\}$ ?