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 $e_1, e_2, \ldots, e_n$ 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 $x_0, x_1, x_2, \ldots, x_n$. 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 [GrYe05] 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 $e_1, e_2, \ldots, e_n$, where e_i is associated with (x_{i-1}, x_i) for $i = 1, 2, \ldots, n$, by its vertex sequence $x_0, x_1, \ldots, x_n$. The notation identifies a path only as far as which the vertices it passes through. There may be more than one path that passes through this sequence of vertices, which will happen if and only if there are multiple edges between two successive vertices in the list.

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 6 in Section 10.1, 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. ◀

Links

EXAMPLE 3 **Paths in Collaboration Graphs** In a collaboration graph, two people a and b are connected by a path when there is a sequence of people starting with a and ending with b such that the endpoints of each edge in the path are people who have collaborated. We will consider two particular collaboration graphs here. First, in the academic collaboration graph of people who have written papers in mathematics, the **Erdős number** of a person m (defined in terms of relations in Supplementary Exercise 14 in Chapter 9) is the length of the shortest path between m and 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.

Links

Replace Kevin Bacon by your own favorite actor to invent a new party game

In the Hollywood graph (see Example 3 in Section 10.1) two actors 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 number of actors can change. In Table 2 we show the number of actors with each Bacon number as of early 2011 using data from the Oracle of Bacon website. The origins of the Bacon number of an actor dates back to the early 1990s, when Kevin Bacon remarked that he had worked with everyone in Hollywood or someone who worked with them. This lead some people to invent a party

TABLE 1 The Number of Mathematicians with a Given Erdős Number (as of early 2006).

Erdős Number	*Number of People*
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

TABLE 2 The Number of Actors with a Given Bacon Number (as of early 2011).

Bacon Number	*Number of People*
0	1
1	2,367
2	242,407
3	785,389
4	200,602
5	14,048
6	1,277
7	114
8	16

game where participants where challenged to find a sequence of movies leading from each actor named to Kevin Bacon. We can find a number similar to a Bacon number using any actor as the center of the acting universe. ◀

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. An undirected graph that is not *connected* is called *disconnected*. We say that we *disconnect* a graph when we remove vertices or edges, or both, to produce a disconnected subgraph.

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

EXAMPLE 4 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 11.

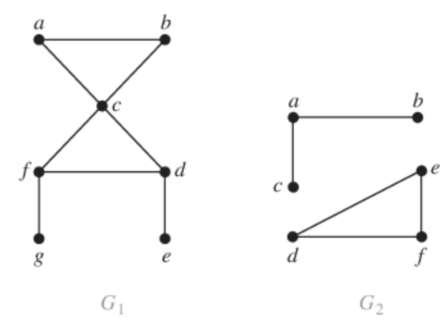


FIGURE 2 **The Graphs G_1 and G_2.**

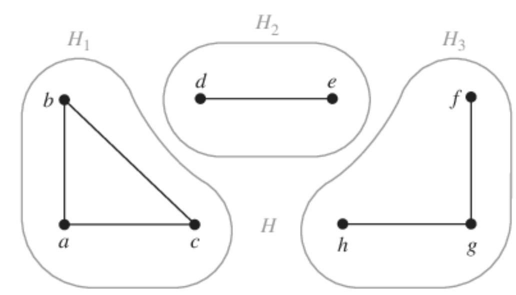


FIGURE 3 **The Graph H and Its Connected Components H_1, H_2, and H_3.**

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 $x_0, x_1, \ldots, x_n$, 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 $0 \le i < j$. This means that there is a path from u to v of shorter length with vertex sequence $x_0, x_1, \ldots, x_{i-1}, x_j, \ldots, x_n$ obtained by deleting the edges corresponding to the vertex sequence $x_i, \ldots, x_{j-1}$. ◁

CONNECTED COMPONENTS 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 5 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. ◀

EXAMPLE 6 **Connected Components of Call Graphs** Two vertices x and y are in the same component of a telephone call graph (see Example 4 in Section 10.1) 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. ◀

How Connected is a Graph?

Suppose that a graph represents a computer network. Knowing that this graph is connected tells us that any two computers on the network can communicate. However, we would also like to understand how reliable this network is. For instance, will it still be possible for all computers to communicate after a router or a communications link fails? To answer this and similar questions, we now develop some new concepts.

Sometimes the removal from a graph of a vertex and all incident edges produces a subgraph with more connected components. 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**. Note that in a graph representing a computer network, a cut vertex and a cut edge represent an essential router and an essential link that cannot fail for all computers to be able to communicate.

EXAMPLE 7 Find the cut vertices and cut edges in the graph G_1 shown in Figure 4.

Solution: The cut vertices of G_1 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_1. ◀

VERTEX CONNECTIVITY Not all graphs have cut vertices. For example, the complete graph K_n, where $n \geq 3$, has no cut vertices. When you remove a vertex from K_n and all edges incident to it, the resulting subgraph is the complete graph K_{n-1}, a connected graph. Connected graphs without cut vertices are called **nonseparable graphs**, and can be thought of as more connected than those with a cut vertex. We can extend this notion by defining a more granulated measure of graph connectivity based on the minimum number of vertices that can be removed to disconnect a graph.

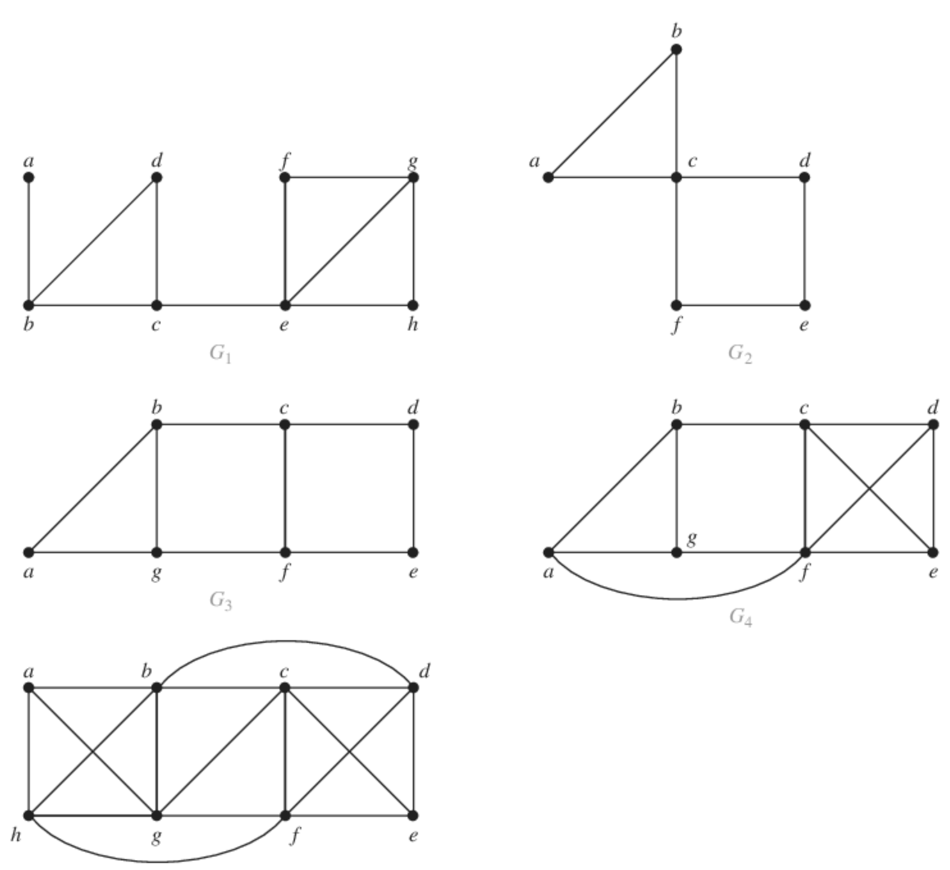


FIGURE 4 **Some Connected Graphs**

A subset V' of the vertex set V of $G = (V, E)$ is a **vertex cut**, or **separating set**, if $G - V'$ is disconnected. For instance, in the graph in Figure 1, the set $\{b, c, e\}$ is a vertex cut with three vertices, as the reader should verify. We leave it to the reader (Exercise 51) to show that every connected graph, except a complete graph, has a vertex cut. We define the **vertex connectivity** of a noncomplete graph G, denoted by $\kappa(G)$, as the minimum number of vertices in a vertex cut.

κ is the lowercase Greek letter kappa.

When G is a complete graph, it has no vertex cuts, because removing any subset of its vertices and all incident edges still leaves a complete graph. Consequently, we cannot define $\kappa(G)$ as the minimum number of vertices in a vertex cut when G is complete. Instead, we set $\kappa(K_n) = n - 1$, the number of vertices needed to be removed to produce a graph with a single vertex.

Consequently, for every graph G, $\kappa(G)$ is minimum number of vertices that can be removed from G to either disconnect G or produce a graph with a single vertex. We have $0 \le \kappa(G) \le n - 1$ if G has n vertices, $\kappa(G) = 0$ if and only if G is disconnected or $G = K_1$, and $\kappa(G) = n - 1$ if and only if G is complete [see Exercise 52(a)].

The larger $\kappa(G)$ is, the more connected we consider G to be. Disconnected graphs and K_1 have $\kappa(G) = 0$, connected graphs with cut vertices and K_2 have $\kappa(G) = 1$, graphs without cut vertices that can be disconnected by removing two vertices and K_3 have $\kappa(G) = 2$, and so on. We say that a graph is ***k*-connected** (or ***k*-vertex-connected**), if $\kappa(G) \ge k$. A graph G is 1-connected if it is connected and not a graph containing a single vertex; a graph is 2-connected, or **biconnected**, if it is nonseparable and has at least three vertices. Note that if G is a k-connected graph, then G is a j-connected graph for all j with $0 \le j \le k$.

EXAMPLE 8 Find the vertex connectivity for each of the graphs in Figure 4.

Solution: Each of the five graphs in Figure 4 is connected and has more than vertex, so each of these graphs has positive vertex connectivity. Because G_1 is a connected graph with a cut vertex, as shown in Example 7, we know that $\kappa(G_1) = 1$. Similarly, $\kappa(G_2) = 1$, because c is a cut vertex of G_2.

The reader should verify that G_3 has no cut vertices. but that $\{b, g\}$ is a vertex cut. Hence, $\kappa(G_3) = 2$. Similarly, because G_4 has a vertex cut of size two, $\{c, f\}$, but no cut vertices. It follows that $\kappa(G_4) = 2$. The reader can verify that G_5 has no vertex cut of size two, but $\{b, c, f\}$ is a vertex cut of G_5. Hence, $\kappa(G_5) = 3$. ◀

EDGE CONNECTIVITY We can also measure the connectivity of a connected graph $G = (V, E)$ in terms of the minimum number of edges that we can remove to disconnect it. If a graph has a cut edge, then we need only remove it to disconnect G. If G does not have a cut edge, we look for the smallest set of edges that can be removed to disconnect it. A set of edges E' is called an **edge cut** of G if the subgraph $G - E'$ is disconnected. The **edge connectivity** of a graph G, denoted by $\lambda(G)$, is the minimum number of edges in an edge cut of G. This defines $\lambda(G)$ for all connected graphs with more than one vertex because it is always possible to disconnect such a graph by removing all edges incident to one of its vertices. Note that $\lambda(G) = 0$ if G is not connected. We also specify that $\lambda(G) = 0$ if G is a graph consisting of a single vertex. It follows that if G is a graph with n vertices, then $0 \le \lambda(G) \le n - 1$. We leave it to the reader [Exercise 52(b)] to show that $\lambda(G) = n - 1$ where G is a graph with n vertices if and only if $G = K_n$, which is equivalent to the statement that $\lambda(G) \le n - 2$ when G is not a complete graph.

λ is the lowercase Greek letter lambda.

EXAMPLE 9 Find the edge connectivity of each of the graphs in Figure 4.

Solution: Each of the five graphs in Figure 4 is connected and has more than one vertex, so we know that all of them have positive edge connectivity. As we saw in Example 7, G_1 has a cut edge, so $\lambda(G_1) = 1$.

The graph G_2 has no cut edges, as the reader should verify, but the removal of the two edges $\{a, b\}$ and $\{a, c\}$ disconnects it. Hence, $\lambda(G_2) = 2$. Similarly, $\lambda(G_3) = 2$, because G_3 has no cut edges, but the removal of the two edges $\{b, c\}$ and $\{f, g\}$ disconnects it.

The reader should verify that the removal of no two edges disconnects G_4, but the removal of the three edges $\{b, c\}$, $\{a, f\}$, and $\{f, g\}$ disconnects it. Hence, $\lambda(G_4) = 3$. Finally, the reader should verify that $\lambda(G_5) = 3$, because the removal of any two of its edges does not disconnect it, but the removal of $\{a, b\}$, $\{a, g\}$, and $\{a, h\}$ does. ◀

AN INEQUALITY FOR VERTEX CONNECTIVITY AND EDGE CONNECTIVITY When $G = (V, E)$ is a noncomplete connected graph with at least three vertices, the minimum degree of a vertex of G is an upper bound for both the vertex connectivity of G and the edge connectivity of G. That is, $\kappa(G) \leq \min_{v \in V} \deg(v)$ and $\lambda(G) \leq \min_{v \in V} \deg(v)$. To see this, observe that deleting all the neighbors of a fixed vertex of minimum degree disconnects G, and deleting all the edges that have a fixed vertex of minimum degree as an endpoint disconnects G.

In Exercise 55, we ask the reader to show that $\kappa(G) \leq \lambda(G)$ when G is a connected noncomplete graph. Note also that $\kappa(K_n) = \lambda(K_n) = \min_{v \in V} \deg(v) = n - 1$ when n is a positive integer and that $\kappa(G) = \lambda(G) = 0$ when G is a disconnected graph. Putting these facts together, establishes that for all graphs G,

$$\kappa(G) \leq \lambda(G) \leq \min_{v \in V} \deg(v).$$

APPLICATIONS OF VERTEX AND EDGE CONNECTIVITY Graph connectivity plays an important role in many problems involving the reliability of networks. For instance, as we mentioned in our introduction of cut vertices and cut edges, we can model a data network using vertices to represent routers and edges to represent links between them. The vertex connectivity of the resulting graph equals the minimum number of routers that disconnect the network when they are out of service. If fewer routers are down, data transmission between every pair of routers is still possible. The edge connectivity represents the minimum number of fiber optic links that can be down to disconnect the network. If fewer links are down, it will still be possible for data to be transmitted between every pair of routers.

We can model a highway network, using vertices to represent highway intersections and edges to represent sections of roads running between intersections. The vertex connectivity of the resulting graph represents the minimum number of intersections that can be closed at a particular time that makes it impossible to travel between every two intersections. If fewer intersections are closed, travel between every pair of intersections is still possible. The edge connectivity represents the minimum number of roads that can be closed to disconnect the highway network. If fewer highways are closed, it will still be possible to travel between any two intersections. Clearly, it would be useful for the highway department to take this information into account when planning road repairs.

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 10 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). ◀

STRONG COMPONENTS OF A DIRECTED GRAPH 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. Note that if a and b are two vertices in a directed graph, their strong components are either the same or disjoint. (We leave the proof of this last fact as Exercise 17.)

EXAMPLE 11 The graph H in Figure 5 has three strongly connected components, consisting of the vertex a; the vertex e; and the subgraph consisting of the vertices b, c, and d and edges (b, c), (c, d), and (d, b). ◀

EXAMPLE 12 **The Strongly Connected Components of the Web Graph** The Web graph introduced in Example 5 of Section 10.1 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 (numbers that have now grown considerably). (See [Br00] for details.)

Links

In 2010 the Web graph was estimated to have at least 55 billion vertices and one trillion edges. This implies that more than 40 TB of computer memory would have been needed to represent its adjacency matrix.

The underlying undirected graph of this Web graph is not connected, but it 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

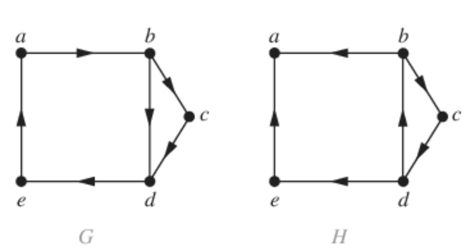


FIGURE 5 **The Directed Graphs G and H.**

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 60.) Example 13 illustrates how this invariant can be used to show that two graphs are not isomorphic.

EXAMPLE 13 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. ◀

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 14 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.

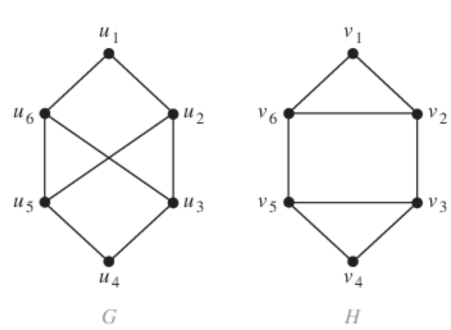


FIGURE 6 **The Graphs G and H.**

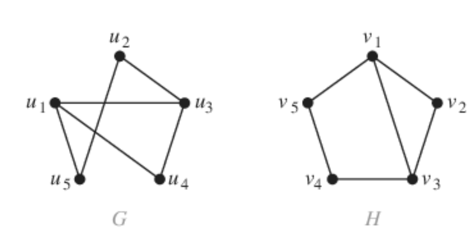


FIGURE 7 **The Graphs G and H.**

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 $\mathbf{A}$ with respect to the ordering $v_1, v_2, \ldots, v_n$ of the vertices of the graph (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 $\mathbf{A}^r$.

Proof: The theorem will be proved using mathematical induction. Let G be a graph with adjacency matrix $\mathbf{A}$ (assuming an ordering $v_1, v_2, \ldots, v_n$ 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 $\mathbf{A}$, because this entry is the number of edges from v_i to v_j.

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

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

where b_{ik} is the (i, k)th entry of $\mathbf{A}^r$. By the inductive 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 15 How many paths of length four are there from a to d in the simple graph G in Figure 8?

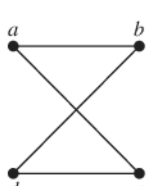


FIGURE 8 The Graph *G*.

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

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

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

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

Extra Examples

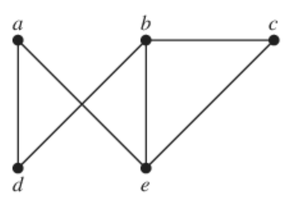
there are exactly eight paths of length four from a to d. By inspection of the graph, we see that a, b, a, b, d; a, b, a, c, d; a, b, d, b, d; a, b, d, c, d; a, c, a, b, d; a, c, a, c, d; a, c, d, b, d; and a, c, d, c, d are the eight paths of length four from a to d. ◀

Theorem 2 can be used to find the length of the shortest path between two vertices of a graph (see Exercise 56), and it can also be used to determine whether a graph is connected (see Exercises 61 and 62).

Exercises

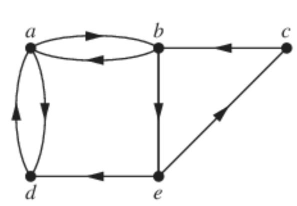
1. Does each of these lists of vertices form a path in the following graph? Which paths are simple? Which are circuits? What are the lengths of those that are paths?

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?

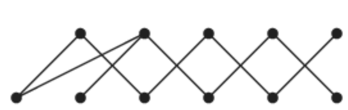
a) a, b, e, c, b **b)** a, d, a, d, a
c) a, d, b, e, a **d)** a, b, e, c, b, d, a



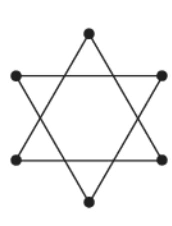
In Exercises 3–5 determine whether the given graph is connected.

3.

4.



5.



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?

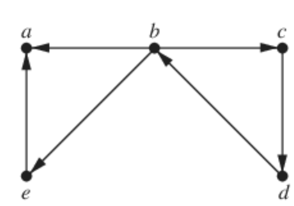
8. What do the connected components of a collaboration graph represent?

9. Explain why in the collaboration graph of mathematicians (see Example 3 in Section 10.1) 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.

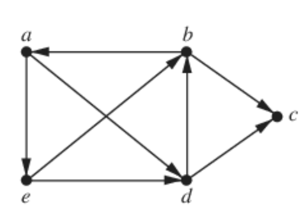
10. In the Hollywood graph (see Example 3 in Section 10.1), when is the vertex representing an actor in the same connected component as the vertex representing Kevin Bacon?

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

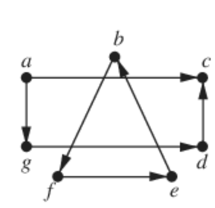
a)



b)

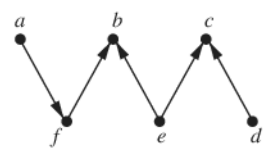


c)

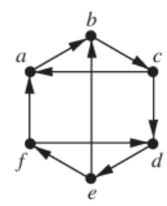


12. Determine whether each of these graphs is strongly connected and if not, whether it is weakly connected.

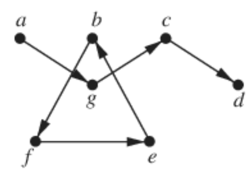
a)



b)



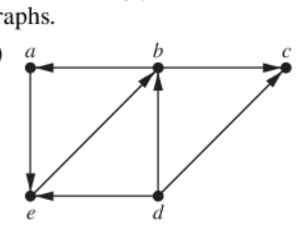
c)



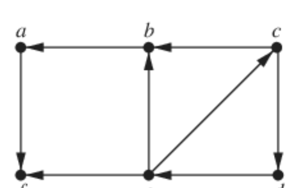
13. What do the strongly connected components of a telephone call graph represent?

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

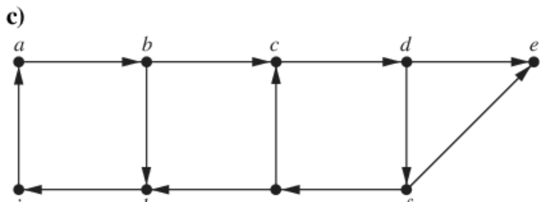
a)



b)

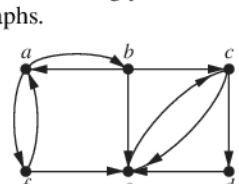


c)

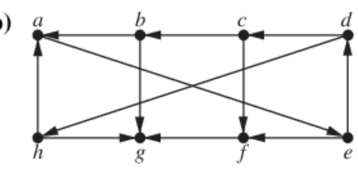


15. Find the strongly connected components of each of these graphs.

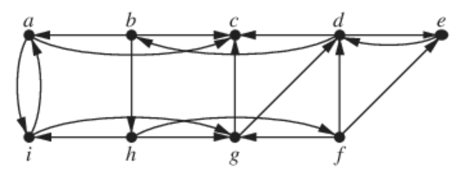
a)



b)



c)



Suppose that $G = (V, E)$ is a directed graph. A vertex $w \in V$ is **reachable** from a vertex $v \in V$ if there is a directed path from v to w. The vertices v and w are **mutually reachable** if there are both a directed path from v to w and a directed path from w to v in G.

16. Show that if $G = (V, E)$ is a directed graph and u, v, and w are vertices in V for which u and v are mutually reachable and v and w are mutually reachable, then u and w are mutually reachable.

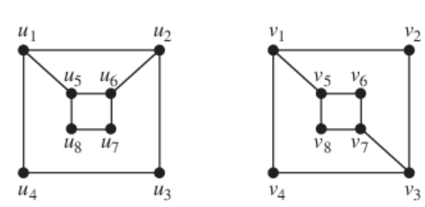
17. Show that if $G = (V, E)$ is a directed graph, then the strong components of two vertices u and v of V are either the same or disjoint. [*Hint:* Use Exercise 16.]

18. 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.

19. Find the number of paths of length n between two different vertices in K_4 if n is

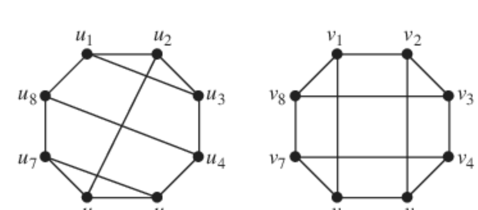
a) 2. **b)** 3. **c)** 4. **d)** 5.

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



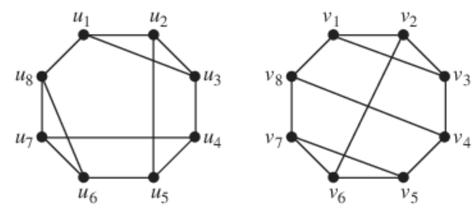
G H

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



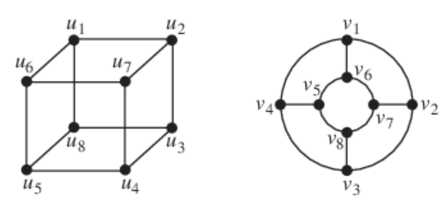
G H

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



G H

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



G H

24. Find the number of paths of length n between any two adjacent vertices in $K_{3,3}$ for the values of n in Exercise 19.

25. Find the number of paths of length n between any two nonadjacent vertices in $K_{3,3}$ for the values of n in Exercise 19.

26. 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.

27. 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.

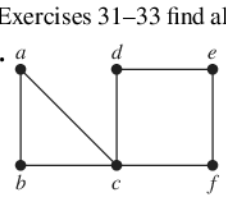
***28.** Show that every connected graph with n vertices has at least $n-1$ edges.

29. 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.

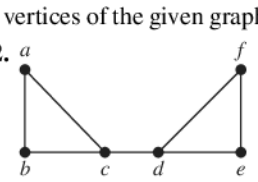
***30.** Show that in every simple graph there is a path from every vertex of odd degree to some other vertex of odd degree.

In Exercises 31–33 find all the cut vertices of the given graph.

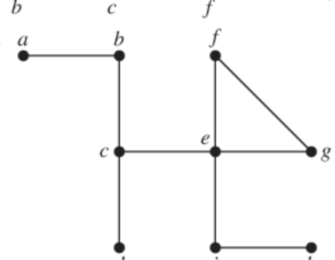
31.



32.



33.



34. Find all the cut edges in the graphs in Exercises 31–33.

***35.** 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.

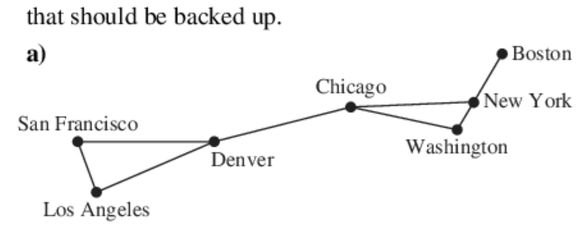
***36.** Show that a vertex c in the connected simple graph G is a cut vertex if and only if there are vertices u and v, both different from c, such that every path between u and v passes through c.

***37.** Show that a simple graph with at least two vertices has at least two vertices that are not cut vertices.

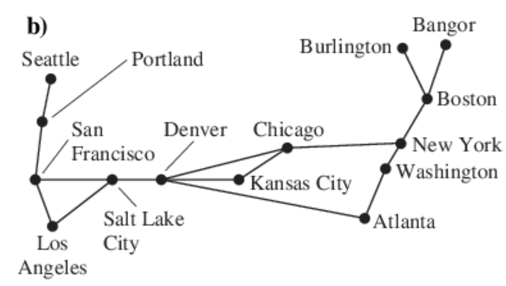
***38.** Show that an edge in a simple graph is a cut edge if and only if this edge is not part of any simple circuit in the graph.

39. 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)



b)



A **vertex basis** in a directed graph G is a minimal set B of vertices of G such that for each vertex v of G not in B there is a path to v from some vertex B.

40. Find a vertex basis for each of the directed graphs in Exercises 7–9 of Section 10.2.

41. What is the significance of a vertex basis in an influence graph (described in Example 2 of Section 10.1)? Find a vertex basis in the influence graph in that example.

42. 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.

***43.** Show that if a simple graph G has k connected components and these components have $n_1, n_2, \ldots, n_k$ vertices, respectively, then the number of edges of G does not exceed

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

***44.** Use Exercise 43 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 ith connected component.]

***45.** Show that a simple graph G with n vertices is connected if it has more than $(n-1)(n-2)/2$ edges.

46. 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.

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

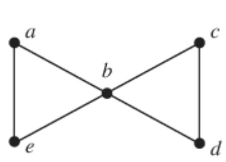
a) 2? **b)** 3? **c)** 4? **d)** 5?

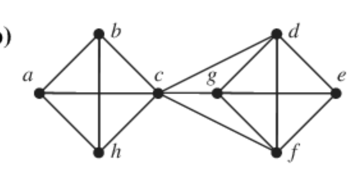
48. Show that each of the following graphs has no cut vertices.

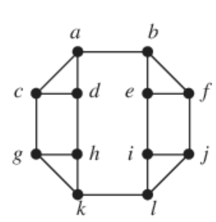
a) C_n where $n \geq 3$
b) W_n where $n \geq 3$
c) $K_{m,n}$ where $m \geq 2$ and $n \geq 2$
d) Q_n where $n \geq 2$

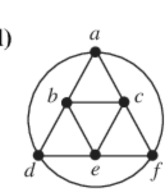
49. Show that each of the graphs in Exercise 48 has no cut edges.

50. For each of these graphs, find $\kappa(G)$, $\lambda(G)$, and $\min_{v \in V} \deg(v)$, and determine which of the two inequalities in $\kappa(G) \leq \lambda(G) \leq \min_{v \in V} \deg(v)$ are strict.

a)


b)


c)


d)


51. Show that if G is a connected graph, then it is possible to remove vertices to disconnect G if and only if G is not a complete graph.

52. Show that if G is a connected graph with n vertices then

a) $\kappa(G) = n - 1$ if and only if $G = K_n$.
b) $\lambda(G) = n - 1$ if and only if $G = K_n$.

53. Find $\kappa(K_{m,n})$ and $\lambda(K_{m,n})$, where m and n are positive integers.

54. Construct a graph G with $\kappa(G) = 1$, $\lambda(G) = 2$, and $\min_{v \in V} \deg(v) = 3$.

***55.** Show that if G is a graph, then $\kappa(G) \leq \lambda(G)$.

56. 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.

57. Use Theorem 2 to find the length of the shortest path between a and f in the graph in Figure 1.

58. Use Theorem 2 to find the length of the shortest path from a to c in the directed graph in Exercise 2.

59. 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.

60. Show that the existence of a simple circuit of length k, where k is an integer greater than 2, is an invariant for graph isomorphism.

61. Explain how Theorem 2 can be used to determine whether a graph is connected.

62. Use Exercise 61 to show that the graph G_1 in Figure 2 is connected whereas the graph G_2 in that figure is not connected.

63. Show that a simple graph G is bipartite if and only if it has no circuits with an odd number of edges.

64. 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?

***65.** Use a graph model and a path in your graph, as in Exercise 64, 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?

66. Suppose that you have a three-gallon jug and a five-gallon jug. You may fill either jug with water, you may empty either jug, and you may transfer water from either jug into the other jug. Use a path in a directed graph 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 jug. Represent these ordered pairs by vertices. Add an edge for each allowable operation with the jugs.]

10.5 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

Links

Only five bridges connect Kaliningrad today. Of these, just two remain from Euler's day.

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 once 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. (For a translation of Euler's original paper see [BiLlWi99].) 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.

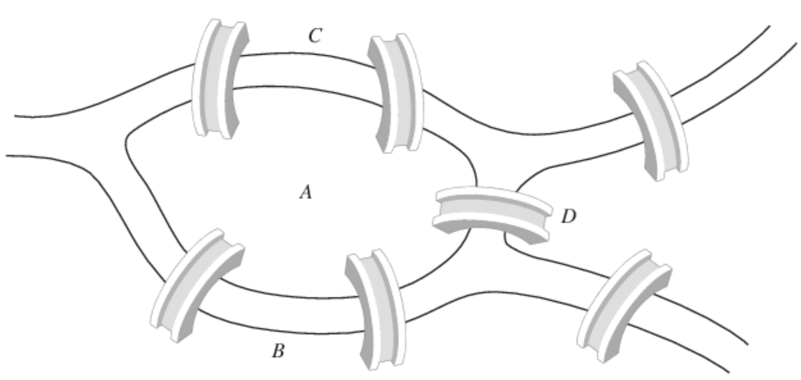


FIGURE 1 **The Seven Bridges of Königsberg.**

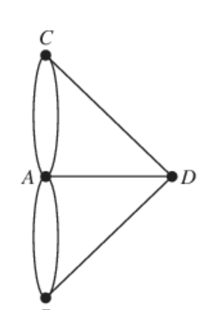


FIGURE 2 **Multigraph Model of the Town of Königsberg.**

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?

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?

Extra Examples

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.

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.

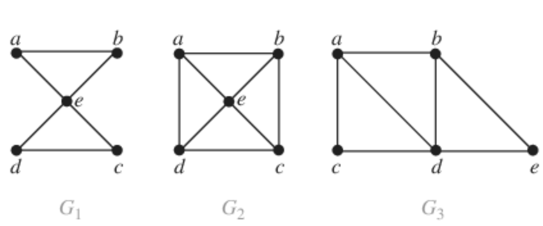


FIGURE 3 The Undirected Graphs G_1, G_2, and G_3.

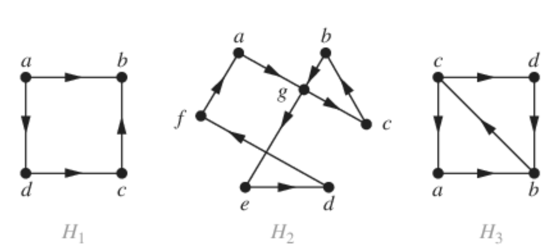


FIGURE 4 The Directed Graphs H_1, H_2, and H_3.

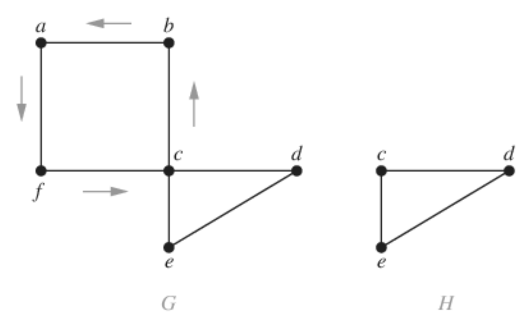


FIGURE 5 Constructing an Euler Circuit in *G*.

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, building it edge by edge. 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 $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$, successively adding edges one by one 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 the 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.)

Links

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.

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 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
    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
return *circuit* {*circuit* is an Euler circuit}

Algorithm 1 provides an efficient algorithm for finding Euler circuits in a connected multigraph G with all vertices of even degree. We leave it to the reader (Exercise 66) to show that the worst case complexity of this algorithm is $O(m)$, where m is the number of edges of G.

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

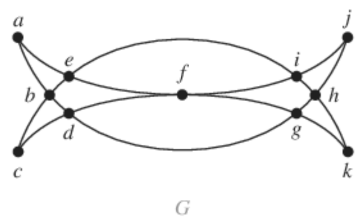


FIGURE 6 **Mohammed's Scimitars.**

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 premble to Exercise 50.

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, 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

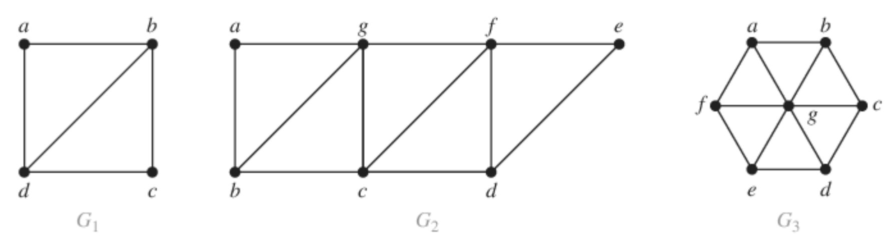


FIGURE 7 **Three Undirected Graphs.**

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 given in Exercises 16 and 17.

APPLICATIONS OF EULER PATHS AND CIRCUITS 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. The problem of finding a circuit in a graph with the fewest edges that traverses every edge at least once 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 the 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 $x_0, x_1, \ldots, x_{n-1}, x_n$ in the graph $G = (V, E)$ is a Hamilton path if $V = \{x_0, x_1, \ldots, x_{n-1}, x_n\}$ and $x_i \neq x_j$ for $0 \leq i < j \leq n$, and the simple circuit $x_0, x_1, \ldots, x_{n-1}, x_n, x_0$ (with $n > 0$) is a Hamilton circuit if $x_0, x_1, \ldots, x_{n-1}, x_n$ is a Hamilton path.

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 string and pegs.

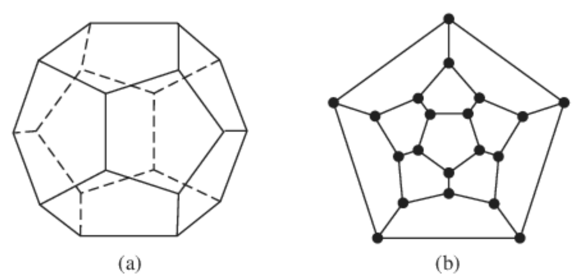


FIGURE 8 Hamilton's "A Voyage Round the World" Puzzle.

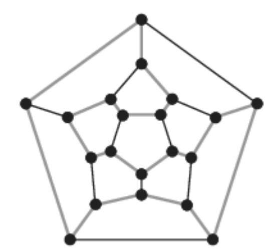


FIGURE 9 A Solution to the "A Voyage Round the World" Puzzle.

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?

Extra Examples

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. ◀

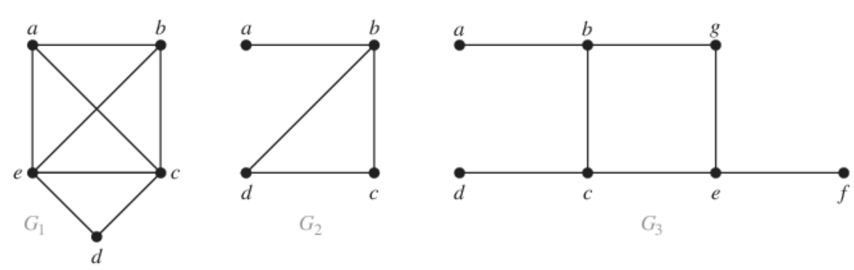


FIGURE 10 Three Simple Graphs.

CONDITIONS FOR THE EXISTENCE OF HAMILTON CIRCUITS 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. 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.

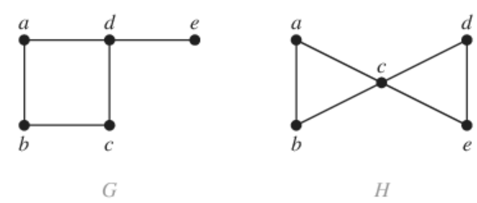


FIGURE 11 **Two Graphs That Do Not Have a Hamilton Circuit.**

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. ◀

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

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.

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 Øystein 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. 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

GABRIEL ANDREW DIRAC (1925–1984) Gabriel Dirac was born in Budapest. He moved to England in 1937 when his mother married the famous physicist and Nobel Laureate Paul Adrien Maurice Dirac, who adopted him. Gabriel A. Dirac entered Cambridge University in 1942, but his studies were interrupted by wartime service in the aviation industry. He obtained his Ph.D. in mathematics in 1951 from the University of London. He held university positions in England, Canada, Austria, Germany, and Denmark, where he spent his last 14 years. Dirac became interested in graph theory early in his career and help raise its status as an important topic of research. He made important contributions to many aspects of graph theory, including graph coloring and Hamilton circuits. Dirac attracted many students to graph theory and was noted as an excellent lecturer.

Dirac was noted for his penetrating mind and held unconventional views on many topics, including politics and social life. Dirac was a man with many interests and held a great passion for fine art. He had a happy family life with his wife Rosemari and his four children.

ØYSTEIN ORE (1899–1968) Ore was born in Kristiania (the old name for Oslo, Norway). In 1922 he received his bachelors degree and in 1925 his Ph.D. in mathematics from Kristiania University, after studies in Germany and in Sweden. In 1927 he was recruited to leave his junior position at Kristiania and join Yale University. He was promoted rapidly at Yale, becoming full professor in 1929 and Sterling Professor in 1931, a position he held until 1968.

Ore made many contributions to number theory, ring theory, lattice theory, graph theory, and probability theory. He was a prolific author of papers and books. His interest in the history of mathematics is reflected in his biographies of Abel and Cardano, and in his popular textbook *Number Theory and its History*. He wrote four books on graph theory in the 1960s.

During and after World War II Ore played a major role supporting his native Norway. In 1947 King Haakon VII of Norway gave him the Knight Order of St. Olaf to recognize these efforts. Ore possessed deep knowledge of painting and sculpture and was an ardent collector of ancient maps. He was married and had two children.

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.

Applications of Hamilton Circuits

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 salesperson problem** or **TSP** (also known in older literature as the **traveling salesman problem**) asks for the shortest route a traveling salesperson 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 10.6.

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 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

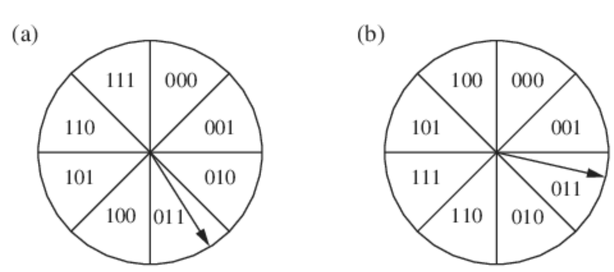


FIGURE 12 **Converting the Position of a Pointer into Digital Form.**

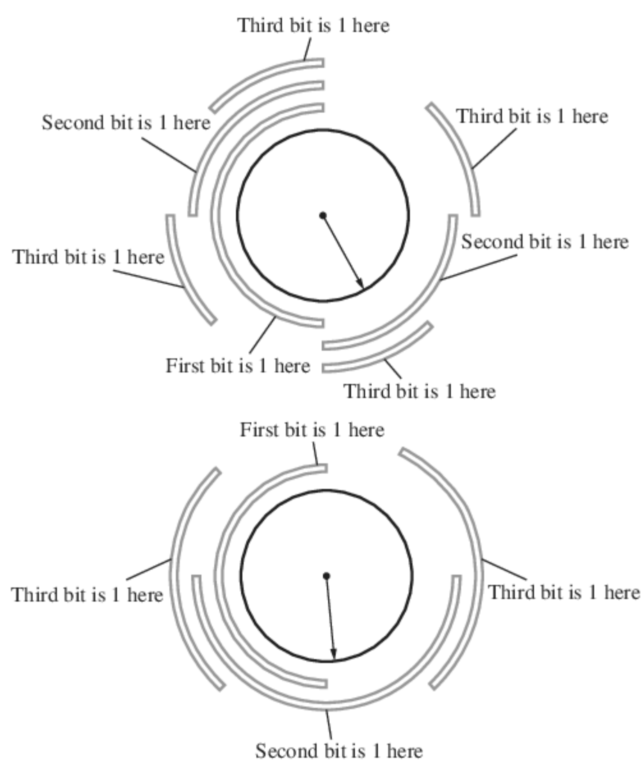


FIGURE 13 **The Digital Representation of the Position of the Pointer.**

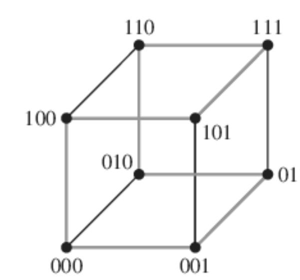


FIGURE 14 **A Hamilton Circuit for Q_3.**

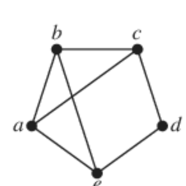
circuit for Q_3 is displayed in Figure 14. The sequence of bit strings differing in exactly one bit produced by this Hamilton circuit is 000, 001, 011, 010, 110, 111, 101, 100.

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. ◀

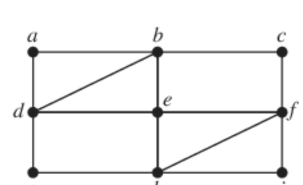
Exercises

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.

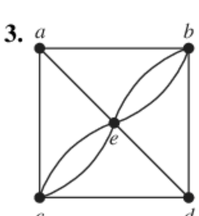
1.



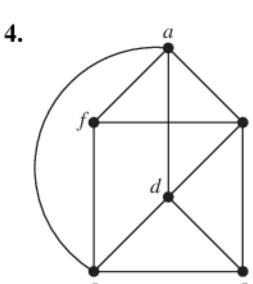
2.



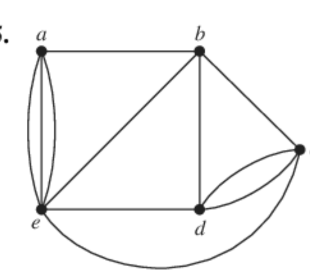
3.



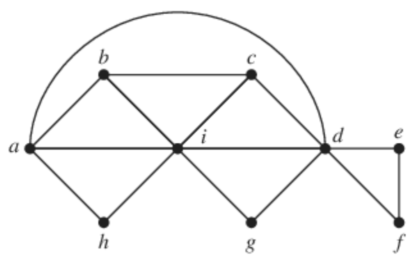
4.



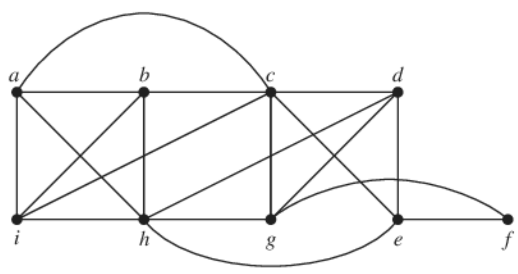
5.



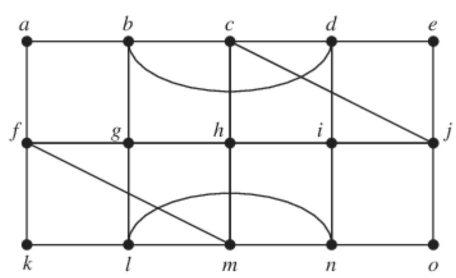
6.



7.

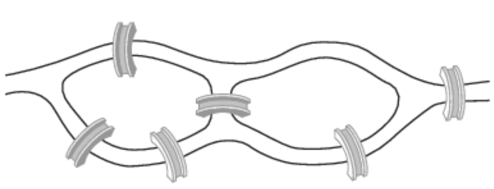


8.



9. Suppose that in addition to the seven bridges of Königsberg (shown in Figure 1) there were two additional bridges, connecting regions *B* and *C* and regions *B* and *D*, respectively. Could someone cross all nine of these bridges 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?



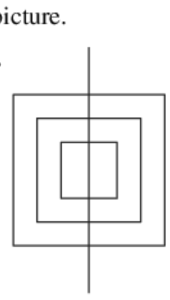
11. When can the centerlines of the streets in a city be painted without traveling a street more than once? (Assume that all the streets are two-way streets.)

12. Devise a procedure, similar to Algorithm 1, for constructing Euler paths in multigraphs.

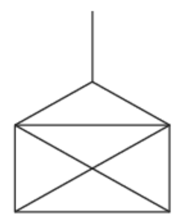
In Exercises 13–15 determine whether the picture shown can be drawn with a pencil in a continuous motion without lifting the pencil or retracing part of the picture.

13.

14.



15.

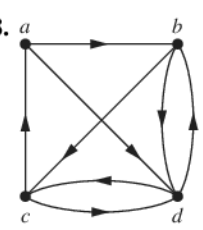


***16.** Show that a directed multigraph having no isolated vertices has an Euler circuit if and only if the graph is weakly connected and the in-degree and out-degree of each vertex are equal.

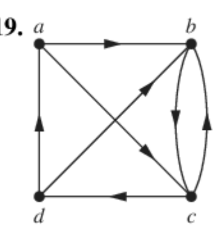
***17.** Show that a directed multigraph having no isolated vertices has an Euler path but not an Euler circuit if and only if the graph is weakly connected and the in-degree and out-degree of each vertex are equal for all but two vertices, one that has in-degree one larger than its out-degree and the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

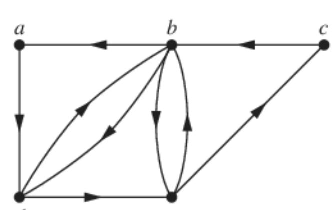
18.



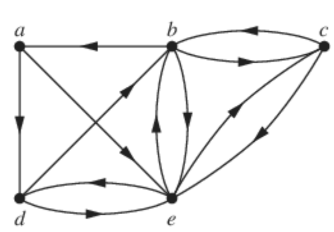
19.



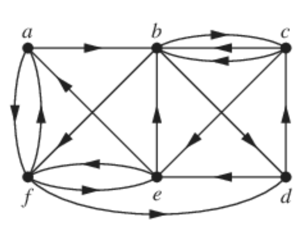
20.



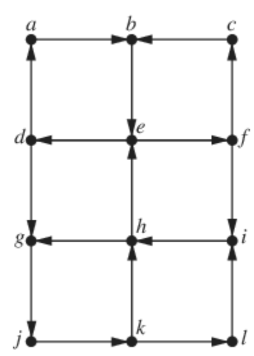
21.



22.



23.



***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?

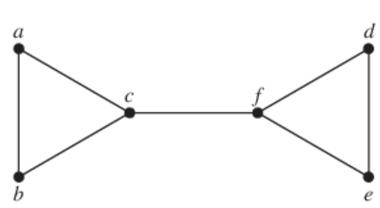
28. For which values of m and n does the complete bipartite graph $K_{m,n}$ have an

a) Euler circuit?
b) Euler path?

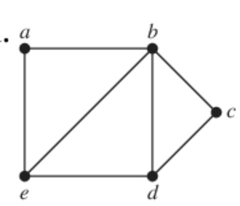
29. Find the least number of times it is necessary to lift a pencil from the paper when drawing each of the graphs in Exercises 1–7 without retracing any part of the graph.

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.

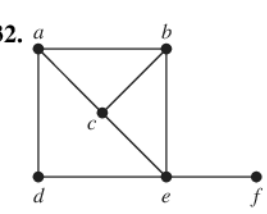
30.



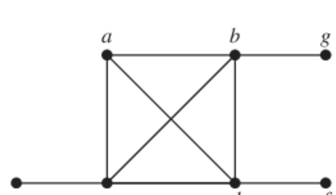
31.



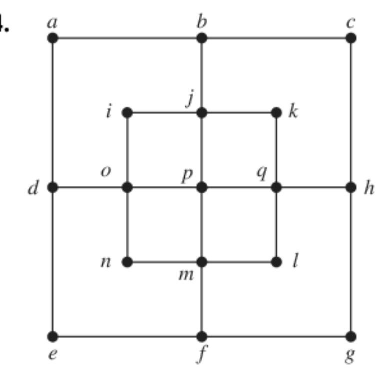
32.



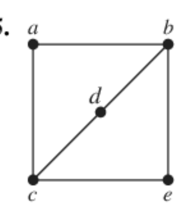
33.



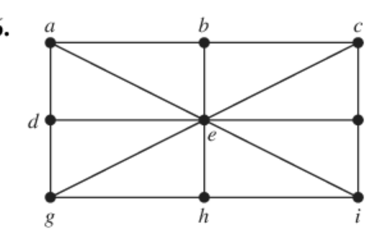
34.



35.



36.



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.

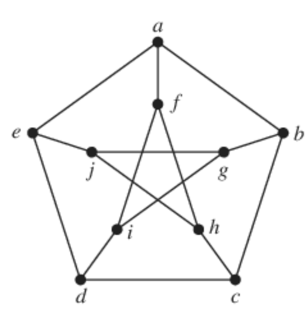
42. Does the graph in Exercise 35 have a Hamilton path? If so, find such a path. If it does not, give an argument to show why no such path exists.

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.

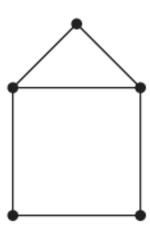
44. For which values of n do the graphs in Exercise 26 have a Hamilton circuit?

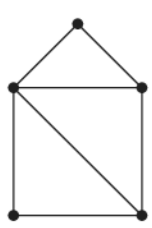
45. For which values of m and n does the complete bipartite graph $K_{m,n}$ have a Hamilton circuit?

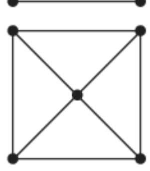
***46.** Show that the **Petersen graph**, shown here, does not have a Hamilton circuit, but that the subgraph obtained by deleting a vertex v, and all edges incident with v, does have a Hamilton circuit.

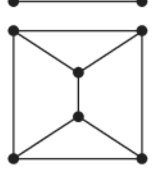


47. For each of these graphs, determine (i) whether Dirac's theorem can be used to show that the graph has a Hamilton circuit, (ii) whether Ore's theorem can be used to show that the graph has a Hamilton circuit, and (iii) whether the graph has a Hamilton circuit.

a)


b)


c)


d)


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, published in 1883, constructs Euler circuits by first choosing an arbitrary vertex of a connected multigraph, and then forming 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.

Links

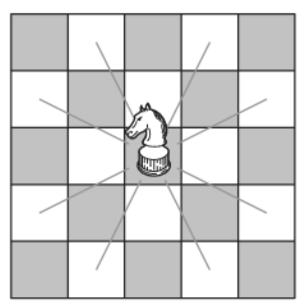
JULIUS PETER CHRISTIAN PETERSEN (1839–1910) Julius Petersen was born in the Danish town of Sorø. His father was a dyer. In 1854 his parents were no longer able to pay for his schooling, so he became an apprentice in an uncle's grocery store. When this uncle died, he left Petersen enough money to return 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 3×3 chessboard.

57. Draw the graph that represents the legal moves of a knight on a 3×4 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 3×4 chessboard.

***60.** Show that there is no knight's tour on a 3×3 chessboard.

***61.** Show that there is no knight's tour on a 4×4 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 8×8 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 $v_1, v_2, \ldots, v_n$ 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 $v_1, v_2, \ldots, v_{k-1}, v_k, v_n, v_{n-1}, \ldots, v_{k+1}, v_1$ is a Hamilton circuit in G. Conclude from this contradiction that Ore's theorem holds.

***66.** Show that the worst case computational complexity of Algorithm 1 for finding Euler circuits in a connected graph with all vertices of even degree is $O(m)$, where m is the number of edges of G.

10.6 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

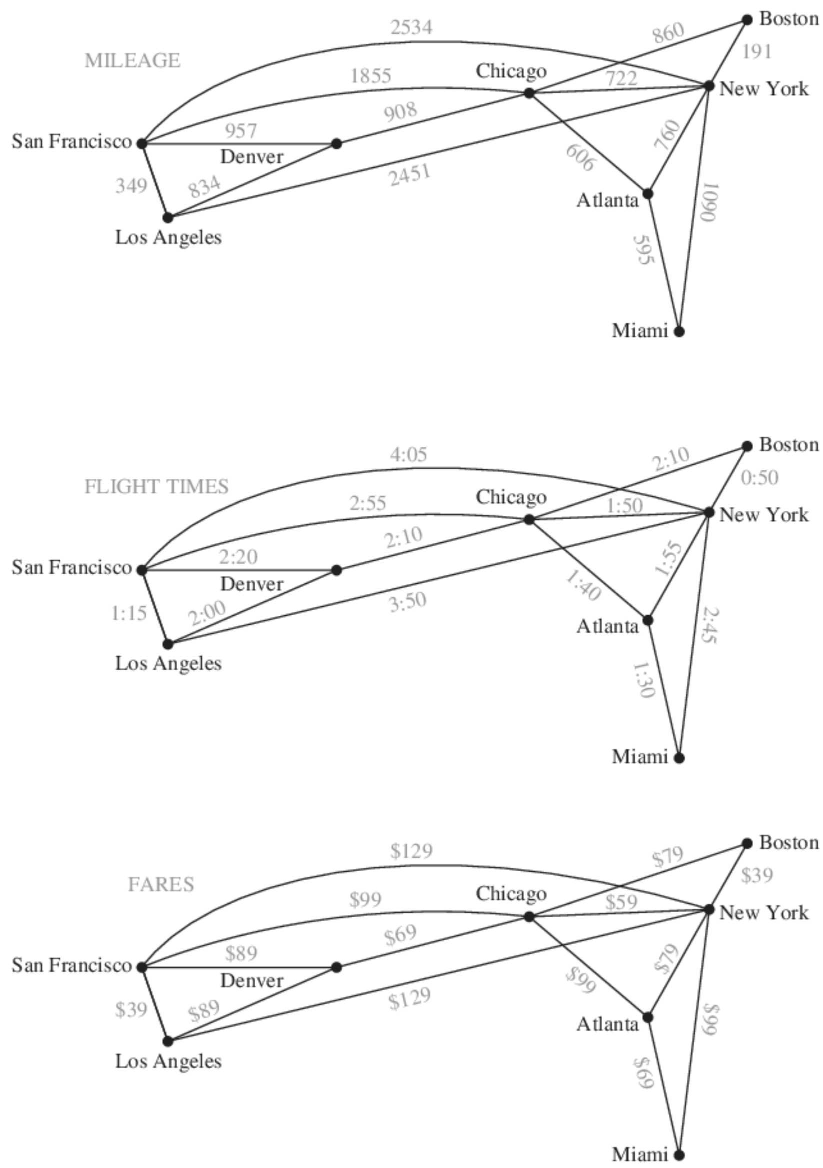


FIGURE 1 **Weighted Graphs Modeling an Airline System.**

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 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

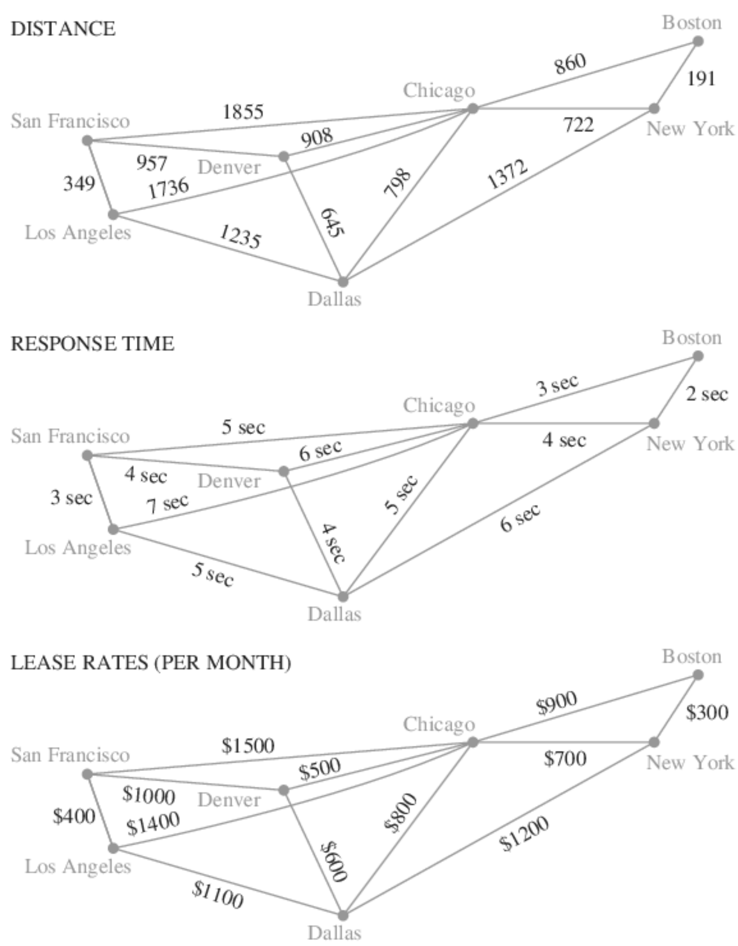


FIGURE 2 **Weighted Graphs Modeling a Computer Network.**

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 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 salesperson problem*, which asks for an order in which a salesperson should visit each of the cities on his route exactly once so that he travels the minimum total distance. We will discuss the traveling salesperson problem later in this section.

A Shortest-Path Algorithm

Links

There are several different algorithms that find a shortest path between two vertices in a weighted graph. We will present a greedy algorithm discovered by the Dutch mathematician Edsger Di-

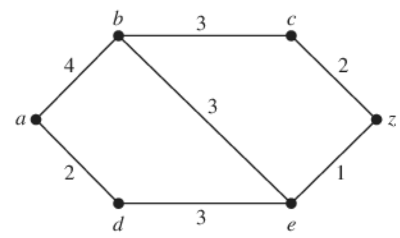


FIGURE 3 **A Weighted Simple Graph.**

jkstra 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 an illustrative example.

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 are formed by adding an edge that has a as one endpoint. These paths have only one edge. They are a, b of length 4 and a, d of length 2. It follows that d is the closest vertex to a, and the shortest path from a to d has length 2.

We can find the second closest vertex by examining all paths that begin with the shortest path from a to a vertex in the set $\{a, d\}$, followed by an edge that has one endpoint in $\{a, d\}$ and its other endpoint not in this set. There are two such paths to consider, a, d, e of length 7 and a, b of length 4. Hence, the second closest vertex to a is b and the shortest path from a to b has length 4.

To find the third closest vertex to a, we need examine only the paths that begin with the shortest path from a to a vertex in the set $\{a, d, b\}$, followed by an edge that has one endpoint in the set $\{a, d, b\}$ and its other endpoint not in this set. There are three such paths, a, b, c of length 7, a, b, e of length 7, and a, d, e of length 5. Because the shortest of these paths is a, d, e, the third closest vertex to a is e and the length of the shortest path from a to e is 5.

Links

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, Austin.

To find the fourth closest vertex to a, we need examine only the paths that begin with the shortest path from a to a vertex in the set $\{a, d, b, e\}$, followed by an edge that has one endpoint in the set $\{a, d, b, e\}$ and its other endpoint not in this set. There are two such paths, a, b, c of length 7 and a, d, e, z of length 6. Because the shorter of these paths is a, d, e, z, the fourth closest vertex to a is z and the length of the shortest path from a 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 a brute force approach by examining the length of every path from a to z. However, this brute force approach is impractical for humans and even for computers for graphs with a large number 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. As a side benefit, this algorithm is easily extended to find the length of the shortest path from a to all other vertices of the graph, and not just to z.

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.

We now give the details of Dijkstra's algorithm. It begins by labeling a with 0 and the other vertices with ∞. 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, ∞ 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 kth 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). Note that the way we choose the vertex u to add to S_k at each step is an optimal choice at each step, making this a greedy algorithm. (We will prove shortly that this greedy algorithm always produces an optimal solution.)

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,

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

where $w(u, v)$ is the length of the edge with u and v as endpoints. 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. Note that we can find the length of the shortest path from a to all other vertices of the graph if we continue this procedure until all vertices are added to the distinguished set.

Demo

ALGORITHM 1 Dijkstra's Algorithm.

procedure *Dijkstra*(G: weighted connected simple graph, with
  all weights positive)
{G has vertices $a = v_0, v_1, \ldots, v_n = z$ and lengths $w(v_i, v_j)$
  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 ∞, and S is the empty set}
while $z \notin S$
  $u :=$ 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}
return $L(z)$ {$L(z) =$ 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.

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 inductive hypothesis the following assertion: At the kth 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 ∞. Hence, the basis case is true.

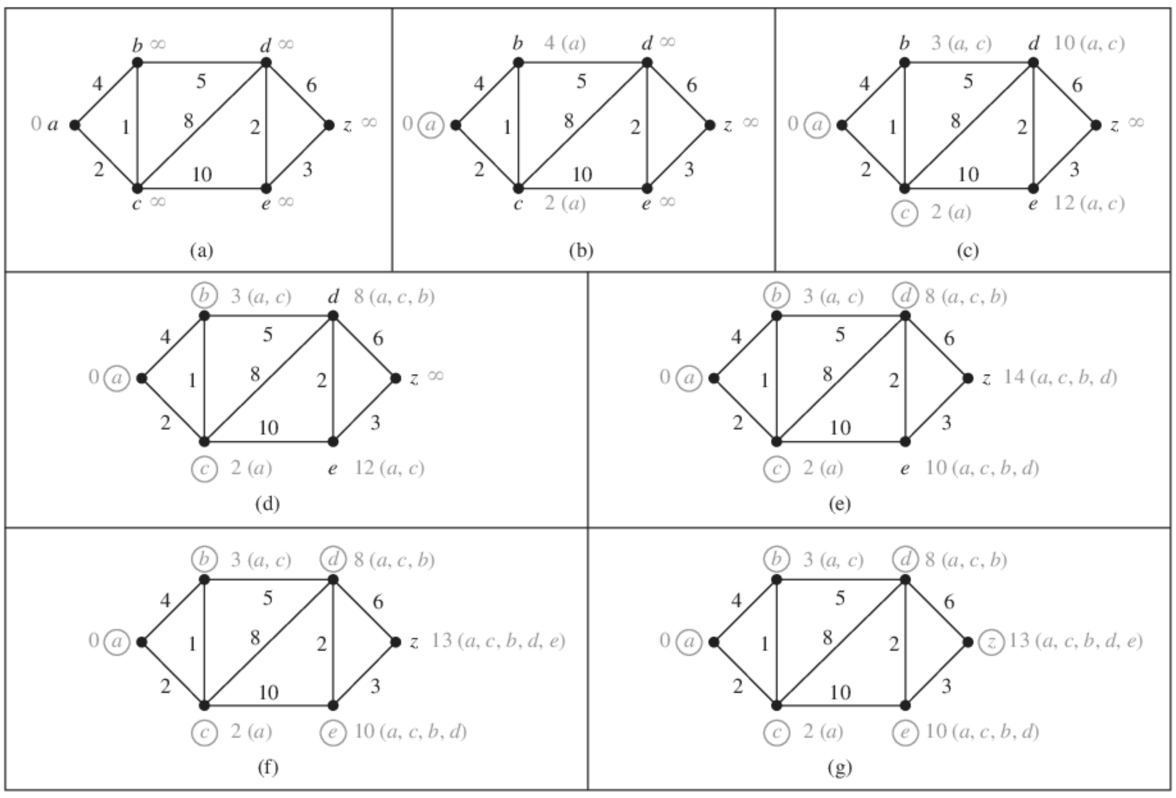


FIGURE 4 **Using Dijkstra's Algorithm to Find a Shortest Path from a to z.**

Assume that the inductive hypothesis holds for the kth 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 kth 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 kth 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 kth 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 $L_{k+1}(u) = \min\{L_k(u), L_k(v) + w(v, u)\}$.

We now state the thereom that we have proved.

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

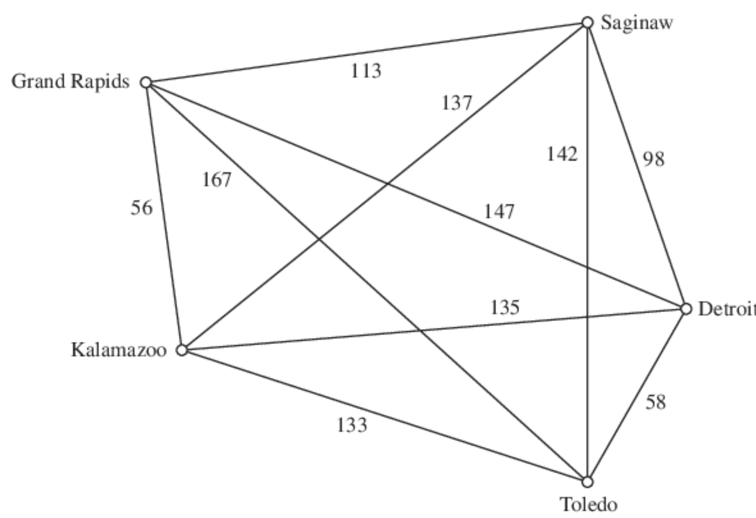


FIGURE 5 **The Graph Showing the Distances between Five Cities.**

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 where n is the number of vertices in the graph, 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 Salesperson Problem

Links

We now discuss an important problem involving weighted graphs. Consider the following problem: A traveling salesperson wants to visit each of n cities exactly once and return to his starting point. For example, suppose that the salesperson 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 salesperson 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).

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 salesperson problem**. The traveling salesperson 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.

An 1832 handbook *Der Handlungsreisende* (The Traveling Salesman) mentions the traveling salesman problem, with sample tours through Germany and Switzerland.

The most straightforward way to solve an instance of the traveling salesperson 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 salesperson 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 3.1×10^{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 salesperson 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 salesperson 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 salesperson 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. (Also, see the preamble to Exercise 46 in the Supplmentary Exercises of Chapter 3.) 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 salesperson 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 salesperson 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

1. For each of these problems about a subway system, describe a weighted graph model that can be used to solve the problem.

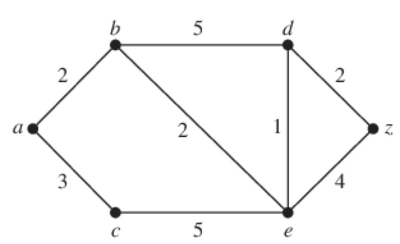
a) What is the least amount of time required to travel between two stops?

b) What is the minimum distance that can be traveled to reach a stop from another stop?

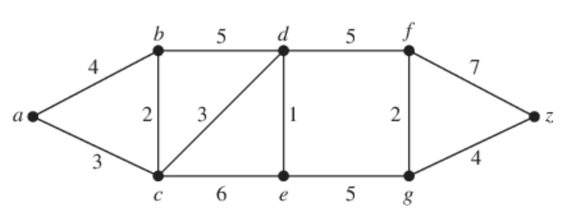
c) What is the least fare required to travel between two stops if fares between stops are added to give the total fare?

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

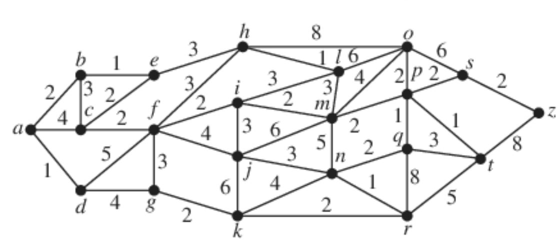
2.



3.



4.



5. Find a shortest path between a and z in each of the weighted graphs in Exercises 2–4.

6. Find the length of a shortest path between these pairs of vertices in the weighted graph in Exercise 3.

a) a and d

b) a and f

c) c and f

d) b and z

7. Find shortest paths in the weighted graph in Exercise 3 between the pairs of vertices in Exercise 6.

8. Find a shortest path (in mileage) between each of the following pairs of cities in the airline system shown in Figure 1.

a) New York and Los Angeles

b) Boston and San Francisco

c) Miami and Denver

d) 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.

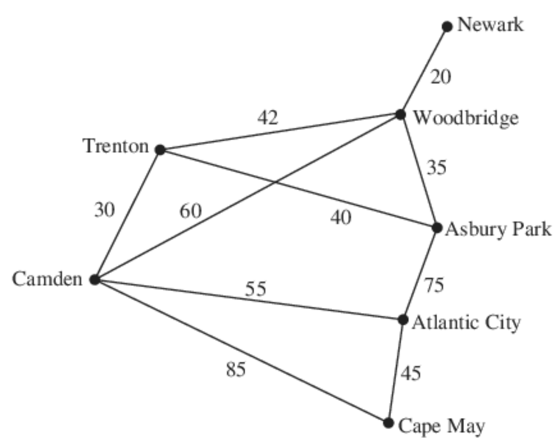
13. Find a least expensive route, in monthly lease charges, between the pairs of computer centers in Exercise 11 using the lease charges given in Figure 2.

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.

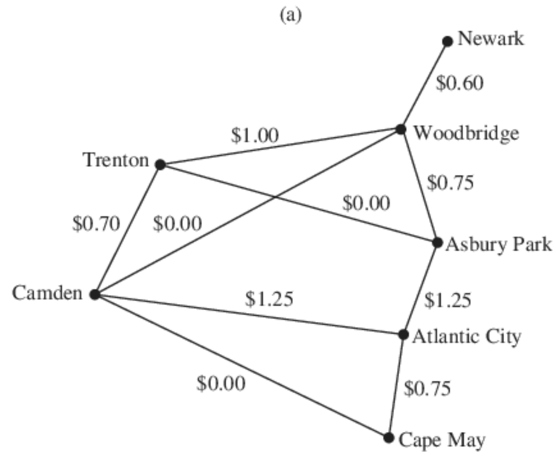
15. Extend Dijkstra's algorithm for finding the length of a shortest path between two vertices in a weighted simple connected graph so that the length of a shortest path between the vertex a and every other vertex of the graph is found.

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)



(b)

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.

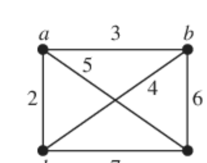
***23.** Give a big-O estimate of the number of operations (comparisons and additions) used by Floyd's algorithm to determine the shortest distance between every pair of vertices in a weighted simple graph with n vertices.

***24.** Show that Dijkstra's algorithm may not work if edges can have negative weights.

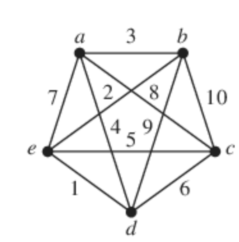
ALGORITHM 2 **Floyd's Algorithm.**

procedure *Floyd*(G: weighted simple graph)
{G has vertices $v_1, v_2, \ldots, v_n$ 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 $d(v_j, v_k) := d(v_j, v_i) + d(v_i, v_k)$
return $[d(v_i, v_j)]$ {$d(v_i, v_j)$ is the length of a shortest path between v_i and v_j for $1 \le i \le n$, $1 \le j \le n$}

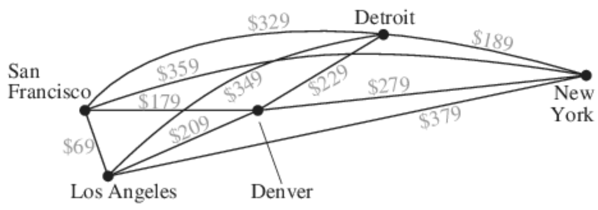
25. Solve the traveling salesperson problem for this graph by finding the total weight of all Hamilton circuits and determining a circuit with minimum total weight.



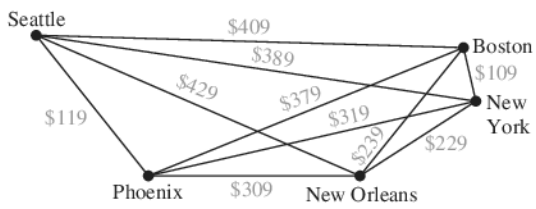
26. Solve the traveling salesperson problem for this graph by finding the total weight of all Hamilton circuits and determining a circuit with minimum total weight.



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.

***31.** The **longest path problem** in a weighted directed graph with no simple circuits asks for a path in this graph such that the sum of its edge weights is a maximum. Devise an algorithm for solving the longest path problem. [*Hint*: First find a topological ordering of the vertices of the graph.]

10.7 Planar Graphs

Introduction

Links

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?

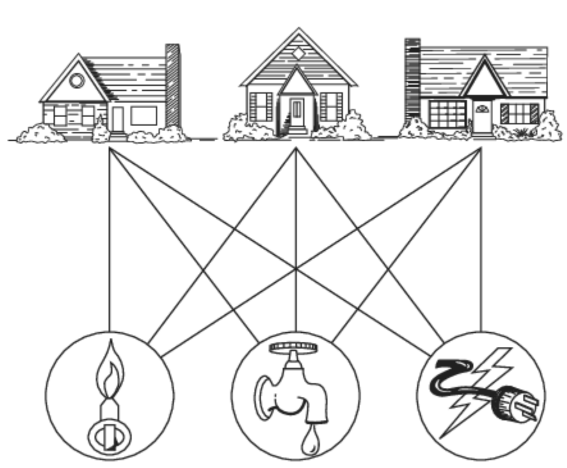


FIGURE 1 **Three Houses and Three Utilities.**

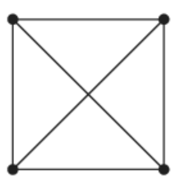


FIGURE 2 **The Graph K_4.**

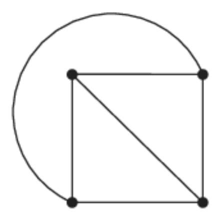


FIGURE 3 **K_4 Drawn with No Crossings.**

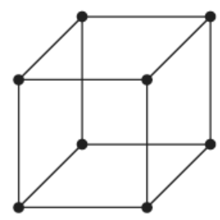


FIGURE 4 **The Graph Q_3.**

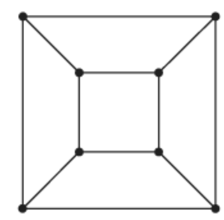


FIGURE 5 **A Planar Representation of Q_3.**

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.

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).

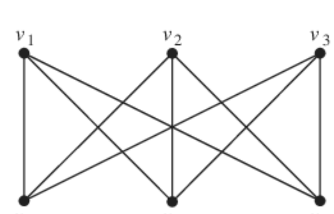


FIGURE 6 **The Graph $K_{3,3}$.**

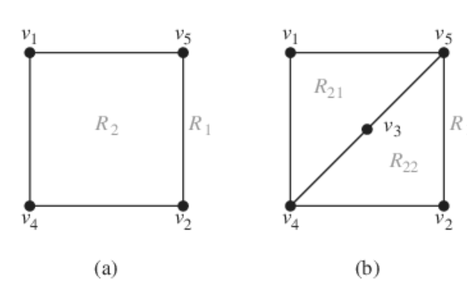


FIGURE 7 **Showing that $K_{3,3}$ Is Nonplanar.**

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). 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.)

APPLICATIONS OF PLANAR GRAPHS 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.)

The planarity of graphs is also useful in the design of road networks. Suppose we want to connect a group of cities by roads. We can model a road network connecting these cities using a simple graph with vertices representing the cities and edges representing the highways connecting them. We can built this road network without using underpasses or overpasses if the resulting graph is planar.

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.

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 $r = e - v + 2$.

Proof: First, we specify a planar representation of G. We will prove the theorem by constructing a sequence of subgraphs $G_1, G_2, \ldots, G_e = G$, 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},

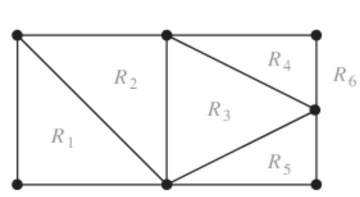


FIGURE 8 **The Regions of the Planar Representation of a Graph.**

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.

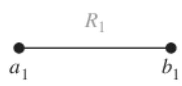


FIGURE 9 The Basis Case of the Proof of Euler's Formula.

Now assume that $r_k = e_k - v_k + 2$. Let $\{a_{k+1}, b_{k+1}\}$ be the edge that is added to G_k to obtain G_{k+1}. There are two possibilities to consider. In the first case, both a_{k+1} and b_{k+1} are already in G_k. These two vertices must be on the boundary of a common region R, or else it would be impossible to add the edge $\{a_{k+1}, b_{k+1}\}$ to G_k without two edges crossing (and G_{k+1} is planar). The addition of this new edge splits R into two regions. Consequently, in this case, $r_{k+1} = r_k + 1$, $e_{k+1} = e_k + 1$, and $v_{k+1} = v_k$. 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, $r_{k+1} = e_{k+1} - v_{k+1} + 2$. 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_k. Suppose that a_{k+1} is in G_k but that b_{k+1} is not. Adding this new edge does not produce any new regions, because b_{k+1} must be in a region that has a_{k+1} on its boundary. Consequently, $r_{k+1} = r_k$. Moreover, $e_{k+1} = e_k + 1$ and $v_{k+1} = v_k + 1$. Each side of the formula relating the number of regions, edges, and vertices remains the same, so the formula is still true. In other words, $r_{k+1} = e_{k+1} - v_{k+1} + 2$. 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

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

◀

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.

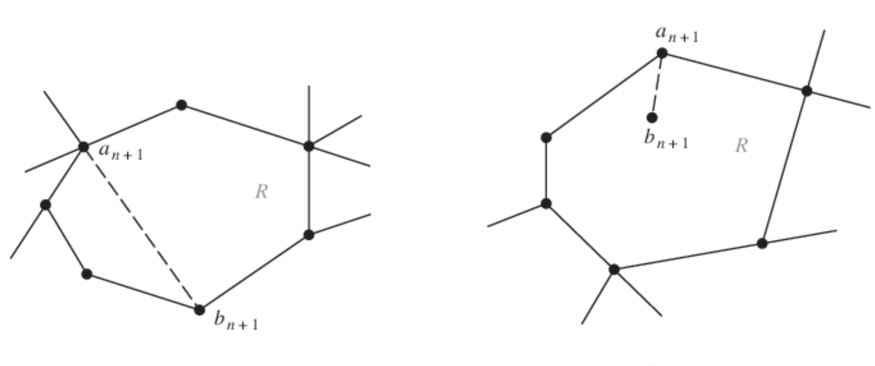


FIGURE 10 Adding an Edge to G_n to Produce G_{n+1}.

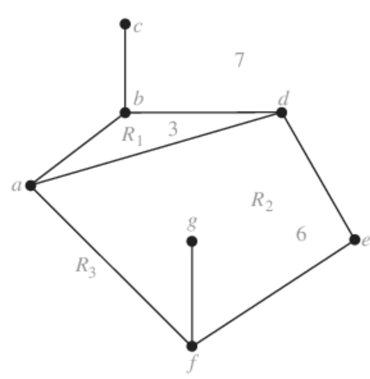


FIGURE 11 **The Degrees of Regions.**

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. We denote the degree of a region R by $\deg(R)$. 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 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,

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

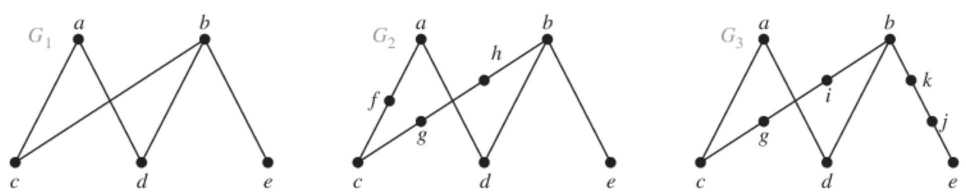


FIGURE 12 **Homeomorphic Graphs.**

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).

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. ◂

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.

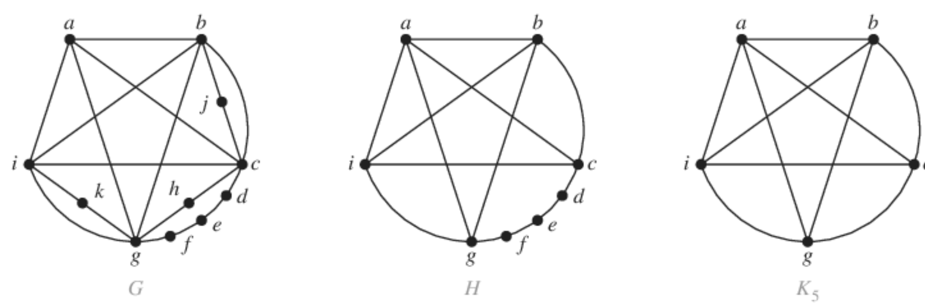


FIGURE 13 **The Undirected Graph G, a Subgraph H Homeomorphic to K_5, and K_5.**

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. ◀

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

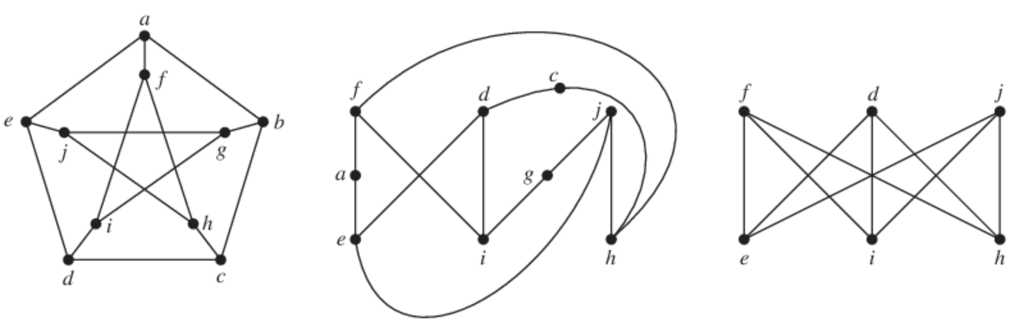
THEOREM 2 A graph is nonplanar if and only if it contains a subgraph homeomorphic to $K_{3,3}$ or K_5.

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.

Extra Examples

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. ◀



(a) (b) H (c) $K_{3,3}$

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

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.)

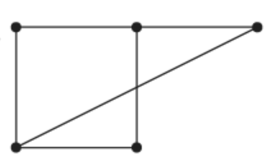
Solution: The subgraph H of the Petersen graph obtained by deleting b and the three edges that have b as an endpoint, shown in Figure 14(b), is homeomorphic to $K_{3,3}$, with vertex sets $\{f, d, j\}$ and $\{e, i, h\}$, because it can be obtained by a sequence of elementary subdivisions, deleting $\{d, h\}$ 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?

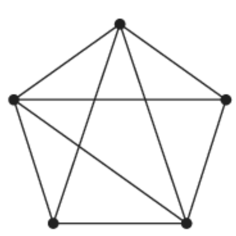
In Exercises 2–4 draw the given planar graph without any crossings.

2.



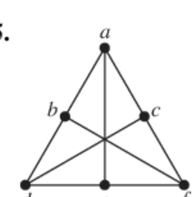
3.

4.

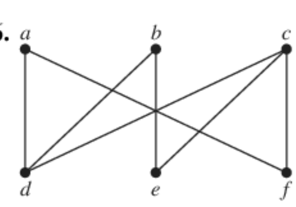


In Exercises 5–9 determine whether the given graph is planar. If so, draw it so that no edges cross.

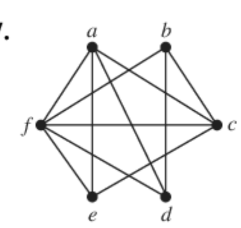
5.



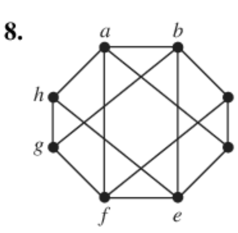
6.



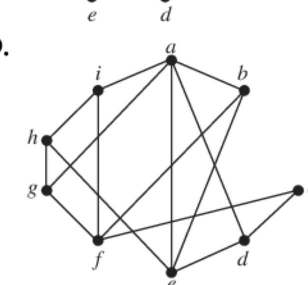
7.



8.



9.



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 $v \geq 4$.

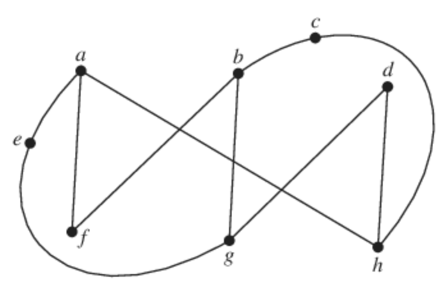
18. Suppose that a planar graph has k connected components, e edges, and v vertices. Also suppose that the plane is divided into r regions by a planar representation of the graph. Find a formula for r in terms of e, v, and k.

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?

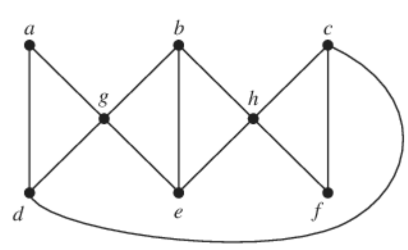
a) K_5 **b)** K_6 **c)** $K_{3,3}$ **d)** $K_{3,4}$

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

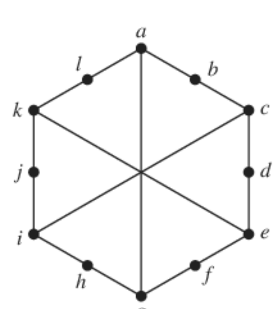
20.



21.

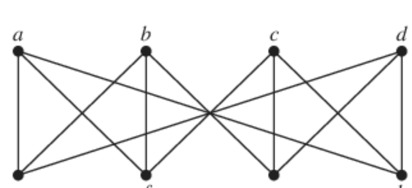


22.

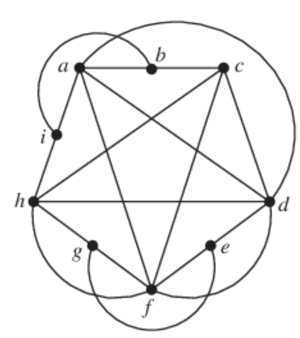


In Exercises 23–25 use Kuratowski's theorem to determine whether the given graph is planar.

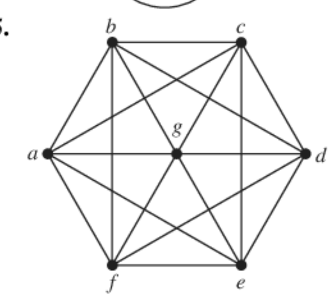
23.



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, where $v \geq 3$, 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, where $v \geq 3$, 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 $K_{m,n}$, where m and n are not both 1, 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.

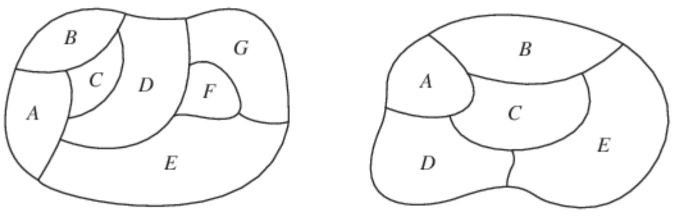


FIGURE 1 **Two Maps.**

10.8 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, 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.

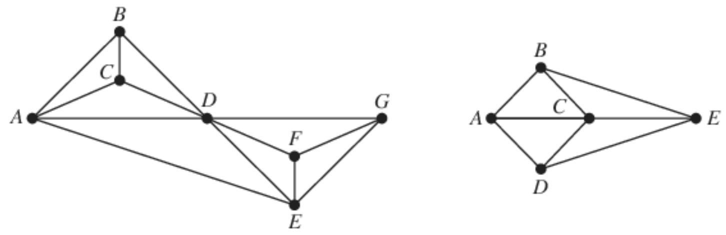


FIGURE 2 **Dual Graphs of the Maps in Figure 1.**

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

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 χ 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? Since their proof appeared, simpler proofs that rely on checking fewer types of possible counterexamples have been found and a proof using an automated proof system has been created. However, no proof not relying on a computer has yet been found.

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.

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.

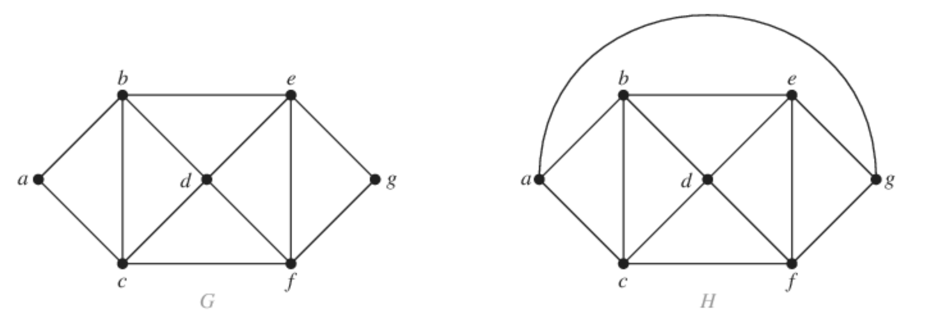


FIGURE 3 **The Simple Graphs G and H.**

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

Extra Examples

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 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. ◀

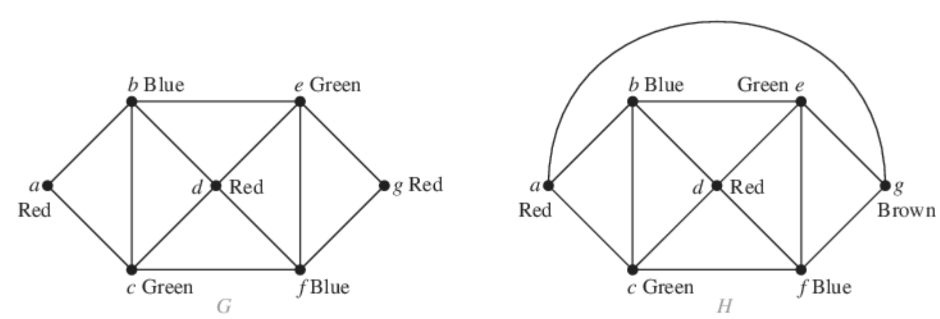


FIGURE 4 **Colorings of the Graphs G and H.**

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.

Links

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.