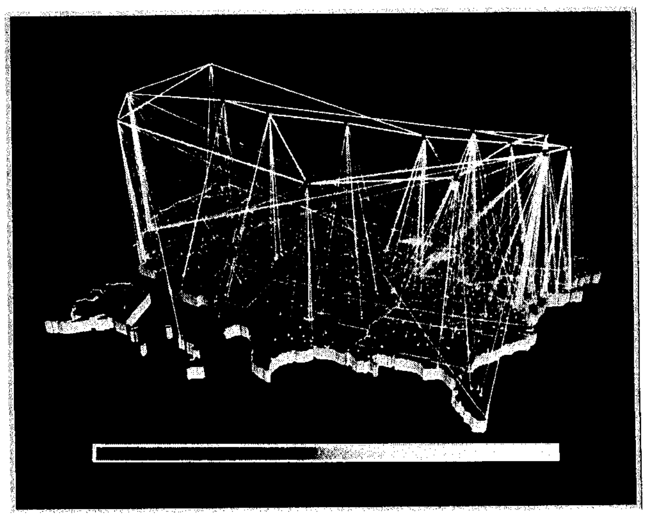
CHAPTER

5

More Graphs, Subgraphs, and Trees

Because of modern technology, elaborate communication networks span the country and most of the earth. These networks allow instant transmission of information between almost any two locations. They affect many aspects of our lives, including the way we work, the way we learn, and the way in which we are entertained.

How does one construct a communication network that links several locations together at the lowest possible cost? How is the most efficient route between two locations in a network found? Can the methods used to find the best route between points in a communication network also be used to plan the best route for an automobile or plane trip? The mathematics of graph theory plays an important role in solving these and many other problems that are important in our ever-changing world.



Lesson 5.1

Planarity and Coloring

In Lesson 4.6, problems involving conflict were solved by modeling them with graphs and then coloring the graphs. The four-color theorem states that any map that can be drawn on the surface of a sphere can be colored with four colors or fewer. If this is true, then why does it take more than four colors to color some graphs?

Explore This

Try to redraw the graphs in Figures 5.1 and 5.2 so that their edges intersect only at the vertices. Try to think of the edges of the graph as rubber bands.

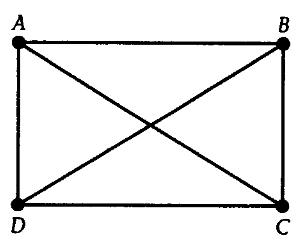


Figure 5.1 K_4 graph.

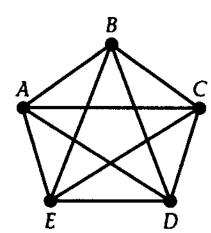


Figure 5.2 K_5 graph.

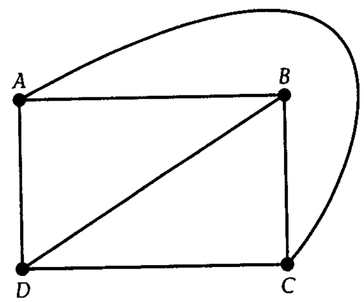


Figure 5.3 Graph in Figure 5.1 redrawn with edges not intersecting.

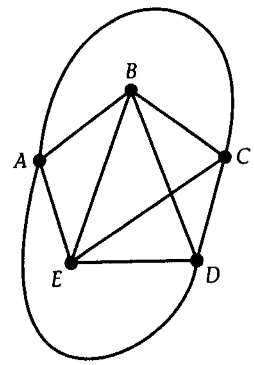


Figure 5.4 An attempt to redraw Figure 5.2 with edges not intersecting.

It is relatively easy to redraw Figure 5.1 so that the edges do not cross (see Figure 5.3), but no matter how hard you try, at least two edges of Figure 5.2 will always intersect (see Figure 5.4). A graph that can be drawn so that no two edges intersect except at a vertex is called a **planar graph**. Figure 5.1 shows a planar graph, and Figure 5.2 shows a graph that is not planar.

When regions of a map are represented by vertices of a graph and edges are drawn between vertices if boundaries exist between regions, the resulting graph is planar. In other words, when a map in a plane or on a sphere is represented by a graph, the resulting graph is always planar. Hence, the four-color theorem can be stated in a different way:

> Every planar graph has a chromatic number that is less than or equal to four.

The question asked earlier about why some graphs require more than four colors can now be answered. Planarity is the key. If a graph is not planar, we do not know how many colors it will take to color it.

One type of graph that is not planar, a K_5, is shown in Figure 5.2. Another nonplanar graph about which many problems have been written is shown in Figure 5.5. Try to redraw it without the edges crossing. Once again you will discover that this is not possible.

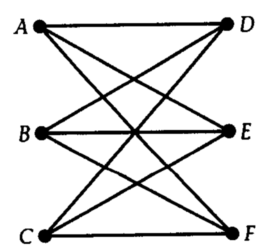


Figure 5.5 $K_{3,3}$ graph.

Bipartite Graphs

The graph in Figure 5.5 has interesting characteristics other than the fact that it is not planar. It is one example of a group of graphs known as **bipartite** graphs. A graph is bipartite if its vertices can be divided into two distinct sets so that each edge of the graph has one vertex in each set. A bipartite graph is said to be complete if it contains all possible edges between the pairs of vertices in the two distinct sets. Complete bipartite graphs can be denoted by $K_{m,n}$, where m and n are the number of vertices in the two distinct sets. Hence, Figure 5.5 is a $K_{3,3}$ graph.

Figure 5.6 is an example of a complete bipartite graph $K_{3,2}$, since its vertices can be separated into two distinct sets {A, B, C} and {X, Y}, every edge has one vertex in each set, and all possible edges from one set to the other are drawn.

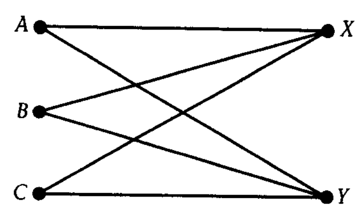


Figure 5.6 $K_{3,2}$ graph.

One way to determine whether a given graph is planar is to try to redraw the graph without edges crossing. For a very large graph, this could prove to be both difficult and time-consuming. In 1930 Kazimierz Kuratowski, a Polish mathematician, provided a partial resolution to this problem of determining the planarity of a graph. He proved that if a graph has a K_5 or $K_{3,3}$ subgraph, it is not planar. A graph G' is said to be a **subgraph** of graph G if all of the vertices and edges of G' are contained in G.

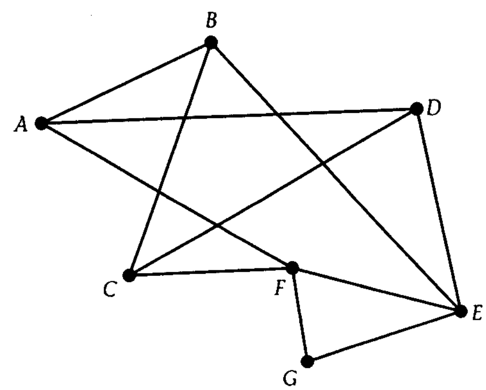
In addition to proving that graphs with K_5 or $K_{3,3}$ subgraphs are not planar, Kuratowski proved that the inverse of his theorem is not true. That is, the lack of a K_5 or $K_{3,3}$ subgraph does not guarantee that the graph is planar (see Exercises 22 and 23 on page 221).

Point of Interest

In practice, approximately 99% of all nonplanar graphs of modest size can be shown to be nonplanar because of a $K_{3,3}$ subgraph rather than a K_5 subgraph.

Example

Determine whether the following graph is planar or nonplanar.

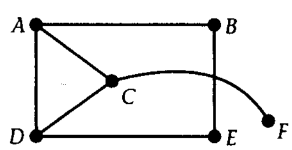


On close inspection of the graph and vertices, *A*, *B*, *C*, *D*, *E*, and *F*, a $K_{3,3}$ subgraph can be found. Therefore, according to Kuratowski's theorem, the graph is nonplanar.

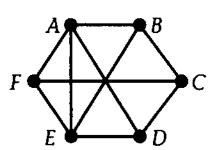
Exercises

In Exercises 1 through 3, decide whether the graph is planar or nonplanar. If the graph is planar, redraw it without edge crossings.

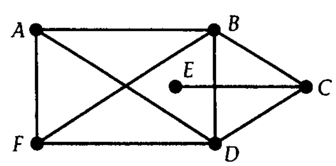
1.



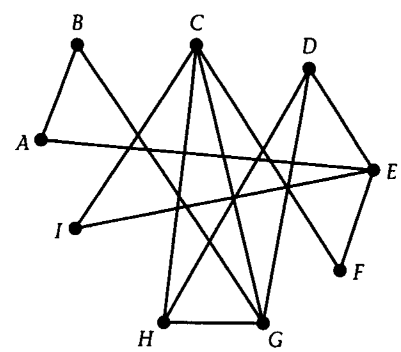
2.



3.

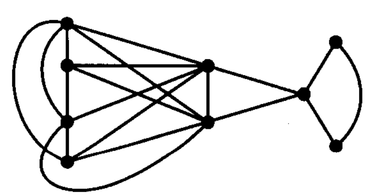


4. The following graph is planar. Draw it without edge crossings.

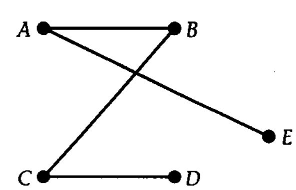


5. By looking at the graph in Exercise 4, how can you tell that it does not contain a K_5 subgraph?

6. Devise a systematic method of searching a graph for a K_5 subgraph. Describe your method in a short paragraph and try it on the following graph. Does the graph contain a K_5 subgraph?



7. The **complement** of a graph G is customarily denoted by $\overline{G}$. The complement $\overline{G}$ has the same vertices as G, but its edges are those not in G. The edges of G and $\overline{G}$ along with vertices from either set would make a complete graph. Draw the complement of the following graph.



8. Every planar graph with nine vertices has a nonplanar complement. Verify this statement for one case by drawing a planar graph with nine vertices and then drawing its complement. For your case, is the complement nonplanar?

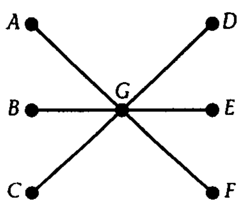
9. The concept of planarity is extremely important to printing circuit boards for the electronics industry. Explain why.

10. Construct the following bipartite graphs.

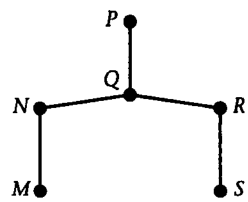
a. $K_{2,3}$ b. $K_{2,4}$

11. For each of the following bipartite graphs, list the two distinct sets into which the vertices can be divided.

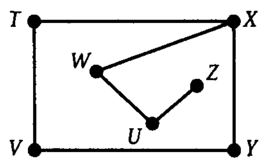
a.



b.

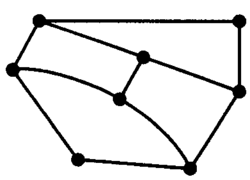


c.

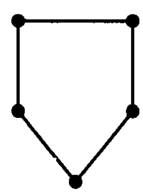


12. State whether the following graphs are bipartite. Explain why or why not.

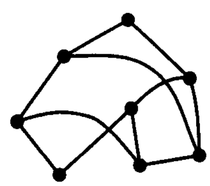
a.



b.



c.



13. Devise a method of telling whether a graph is bipartite. Write a short algorithm for your method.

14. How many edges are in a $K_{2,3}$ graph? A $K_{4,3}$? A $K_{m,n}$?

15. When does a bipartite graph $K_{m,n}$ have an Euler circuit?

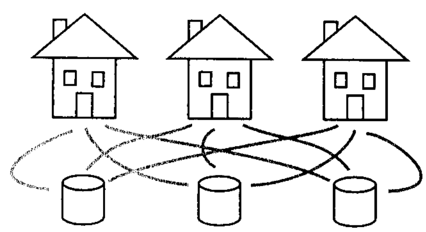
16. What is the chromatic number of a $K_{m,n}$ graph?

17. At Ms. Johnson's party, six men and five women walk into the dining room. If each man shakes hands with each woman, how many handshakes will occur? Represent this situation with a graph. What kind of a graph is it?

18. Describe a situation that can be represented by a bipartite graph that is not complete.

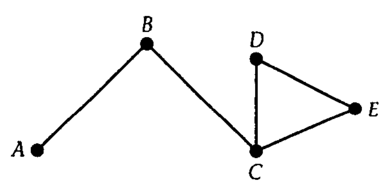
19. The following puzzle is often referred to as the *Wells and Houses problem* or the *Utilities problem*.

Three houses and three wells are built on a piece of land in an arid country. Because it seldom rains, the wells often run dry, and so each house must have access to each well. Unfortunately, the occupants of the three houses dislike one another and want to construct paths to the wells so that no two paths cross.



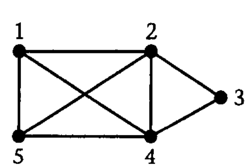
Draw a graph to illustrate this problem. Is it possible to satisfy the wishes of the feuding families? Explain why or why not.

20.

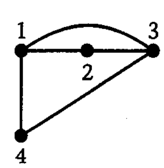


The preceding graph is a subgraph of which following graph(s). Explain how you know.

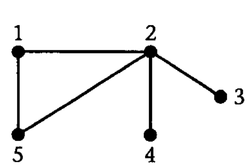
a.



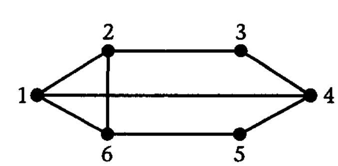
b.



c.



d.



21. Kuratowski proved the following conditional statement.

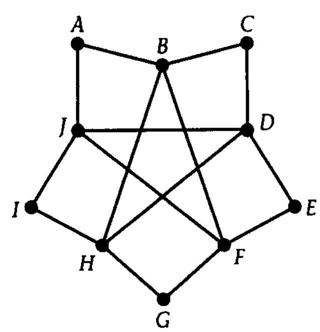
> If a graph contains a K_5 or a $K_{3,3}$ subgraph, then it is not planar.

He also proved that the inverse of the statement is false:

> If a graph does not contain a K_5 or a $K_{3,3}$ subgraph, then it is planar.

State the converse of the conditional statement. Do you think that it is true or false?

22. Is the following graph planar? Does it contain a K_5 or a $K_{3,3}$ as a subgraph?



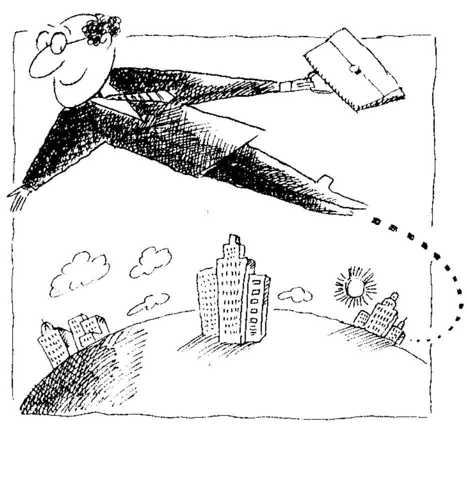
23. For many years mathematicians thought that all nonplanar graphs had either a K_5 or a $K_{3,3}$ as a subgraph. Then nonplanar graphs such as the one in Exercise 22 were found. The graph in Exercise 22 is said to be an **extension** of a K_5 because it was formed by adding a vertex or vertices to the edges of the K_5 graph. Extensions of K_5 and $K_{3,3}$ graphs are nonplanar. This discovery shows that the converse of Kuratowski's theorem is false (Exercise 21).

Redraw the graph in Exercise 22 to show that it is an extension of a K_5.

Lesson 5.2

The Traveling Salesperson Problem

In Lesson 4.5, you explored circuits that visited each vertex of your graph exactly once (Hamiltonian circuits). In this lesson, you will extend your thinking on this original problem and examine a type of problem known as a **traveling salesperson problem** (TSP). These TSPs involve finding a Hamiltonian circuit of minimum value such as time, distance, or cost. Optimization problems of this type are becoming increasingly important in the world of communications, warehousing, airlines networking, road networking, and building wiring.



Explore This

Suppose you are a salesperson who lives in St. Louis. Once a week you have to travel to Minneapolis, Chicago, and New Orleans and then return home to St. Louis. The graph in Figure 5.7 represents the trips that are available to you. The edges of the graph are labeled with the cost of each possible trip. For instance, the cost of making a trip from Chicago to New Orleans is $910. When each edge of a graph is assigned a number (weight), the graph is called a **weighted graph**. The numbers associated with the edges often represent such things as distance, cost, or time.

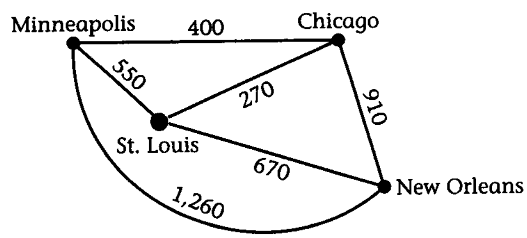


Figure 5.7 Graph of four cities with the costs of traveling between them.

Since you own your own business, it is important that you minimize travel costs. To help save money, find the least expensive route that begins in St. Louis, visits each of the other cities exactly once, and returns to St. Louis.

One way to solve the problem of finding the least expensive route in Figure 5.7 is to list every possible circuit, along with its cost. A tree diagram like the one in Figure 5.8 is helpful in organizing the possibilities. Inspection of all possible routes shows that the optimal solution is the circuit *S, M, C, N, S* or the circuit in reverse order *S, N, C, M, S*.

Solving the TSP for four vertices is not too difficult or time-consuming because only six possibilities need to be considered, but as the number of vertices increases, so does the number of possible circuits. Hence, checking

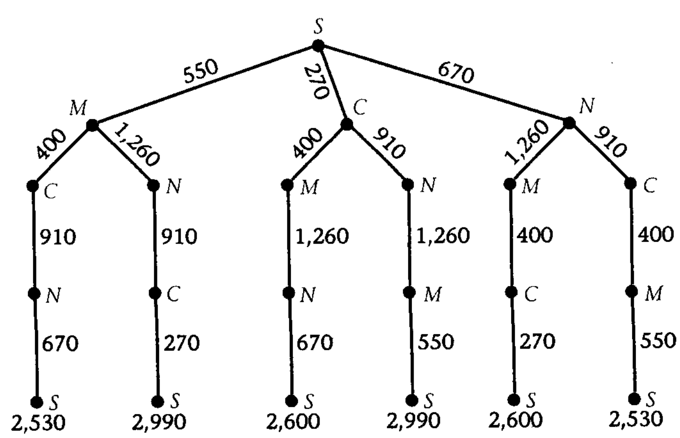


Figure 5.8 Tree diagram of every possible circuit from St. Louis to each of the other cities and back to St. Louis.

all possibilities soon becomes impractical, if not impossible. Even with the help of a computer that can do computations at the rate of 1 billion per second, it would take more than 19 million years for the computer the find the weights of every circuit for a graph with 25 vertices!

More Efficient Algorithms

Is it possible that a faster, more efficient algorithm exists? You might try, for example, to begin at a vertex, look for the vertex nearest to your starting vertex, move to it, and continue until you complete the circuit.

To explore this method for the graph in Figure 5.7 on page 223, begin at St. Louis, move to the nearest neighboring vertex (Chicago), then to the nearest vertex not yet visited, and return to St. Louis when you have visited all of the other cities (270 + 400 + 1,260 + 670 = 2,600). This procedure is known as the **nearest-neighbor algorithm**. Although the solution was reached quickly, notice that in this case, it is not the best possible one. A method such as this, which produces a quick and reasonably close-to-optimal solution, is known as a **heuristic method**.

The choice of method now becomes a trade-off. The method of inspecting every possible path guarantees the best route but is prohibitively slow. The nearest-neighbor method is quick but does not necessarily produce the optimal solution. There is no known computationally efficient method of solving all traveling salesperson problems. But with the

Using Computers and Theories of Evolution to Solve Problems

INDIANAPOLIS STAR AND NEWS
by Eric B. Schoch

Indianapolis—When nature has a problem to solve, it uses natural selection, genetics, mutations and other tools of evolution to give us everything from Mozart to microbes.

The International Conference on Evolutionary Computations brought together about 220 researchers—more than half of them from outside North America—to discuss software that uses the rules of biology to work more intelligently.

Consider the traveling salesman problem. Even for a computer, this trial-and-error system could take a long time.

The alternative is to turn the problem over to computerized evolution: Let the computer create a bunch of possible routes at random, then rank them. Have the computer reproduce the routes, giving the better ones more chances of reproducing—that's survival of the fittest. During the reproduction process, sometimes have the routes swap pieces, the way chromosomes sometimes swap sections when cells divide. Make some random tiny changes in some routes, and call them mutations.

Over time, as generation follows generation, a good answer to the route problem will evolve.

discovery of more and more efficient algorithms, hope has increased that better solutions will be forthcoming. These solutions to TSPs are of great interest because they translate into savings of millions of dollars for certain areas of the economy.

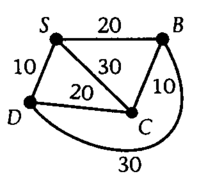
The current record-setting solution for a TSP was set in May 1998 by Robert Bixby and David Applegate of Rice University and their colleagues. This 13,509-node problem consisted of all cities in the United States with populations ≥ 500 and is one of the problems from a library of TSP problems set up by Gerhard Reinelt, a professor at the University of Heidelberg, to provide a standard set of test problems. According to Applegate, "the story isn't over—there is still work to be done, and there are a few smaller problems which are still unsolved."

Exercises

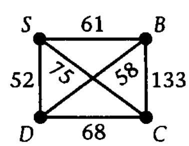
In Exercises 1 through 4,

a. Construct a tree diagram showing all possible circuits that begin at *S*, visit each vertex of the graph exactly once, and end at *S*.
b. Find the total weight of each route.
c. Identify the shortest circuit.
d. Use the nearest-neighbor algorithm to find the shortest circuit.
e. Does the nearest-neighbor algorithm produce the optimal solution?

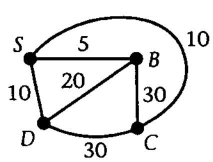
1.



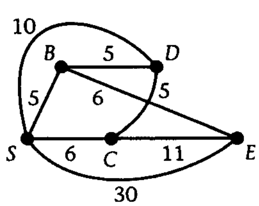
2.



3.



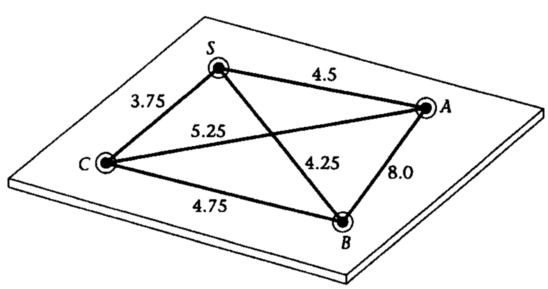
4.



5. In a graph with 10 vertices, 9!, or $9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$, possible Hamiltonian circuits exist if the beginning vertex is known.
   a. Assume that a computer can perform calculations at the rate of 1 million per second. About how long will it take the computer to check 9! possibilities? What if the graph had 15 vertices (14! possible circuits)?
   b. According to October 29, 1998 news reports, a computer now exists that can do 1 trillion computations per second! How long will it take this new computer to check a graph with 9! possible circuits? With 14! possible circuits? With 20! possible circuits?

6. Give two examples of a situation in which a solution to the traveling salesperson problem would be beneficial.

7. The following figure shows a circuit board and the distances in millimeters between holes that must be drilled by a drilling machine. Since it is advantageous in terms of time to minimize the distance traveled, find the shortest possible circuit for the machine to travel and the total distance for that circuit. (Assume the machine has to begin and end at point S.)



Point of Interest

Mathematicians in IBM's Tokyo Research Laboratory have developed a drill route optimization system that has a TSP approximation algorithm as its core. The new system reduces the drill route length on a circuit board by an average of 80% and the operation time by 15% on the average.

Projects

8. Create a TSP poster that illustrates the use of the nearest-neighbor algorithm. At a minimum, your poster should include the following elements.

- A map of your neighborhood, town, or state with five or six selected locations highlighted.
- A table that shows the distances between each pair of selected locations.
- A weighted graph that models the situation.
- An explanation of how the algorithm is used to find the minimum distance from one selected starting location to each of the others and returns to the original starting point.

9. Select five colleges or universities that you might be interested in attending. After researching the schools of choice, write a paper that incorporates the following.

- A short paragraph on each school explaining your interest.
- A weighted, complete graph that shows the five schools and your home.
- An optimal route based on the shortest path that starts at your home, visits each of the five schools, and returns to your home.

Car of Future: Virtual Design and Intelligent Navigation

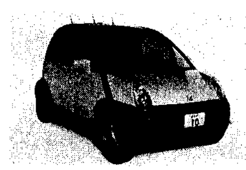
Christian Science Monitor,
August 23, 1996,
by Paul A. Eisenstein

The sleek two-seater merges into the dense freeway traffic, nudging into the express lane.

Settling into the flow of Monday morning rush hour, the driver slips her hands off the steering wheel, pulls out the morning paper and settles back to read as her car races along at 100 miles per hour.

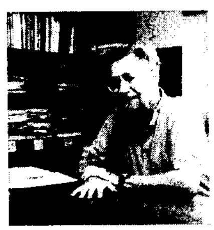
A scenario for disaster? Certainly on today's highways. But by the mid-21st century, this may be a perfectly common sight. Or so goes the vision of highway and automotive planners in the United States.

The federal government has authorized a national smart-car program that could be in place within a decade. It would create an intelligent highway system in as many as 75 major US cities. They're likely to look a lot like ADVANCE. Vehicles participating in the program are equipped with on-board navigation computers. Punch in a destination, and the system automatically plots out the best route. If there's a tie-up along the way, the regional center radios an alert, and the car's computer plots an alternate route.



ONE-SEAT WONDER: Honda Motor Company developed this prototype single-seat car for commuting. The hybrid-engine ICVS-1 was designed as part of a government-aided 'intelligent community vehicle system' in Japan.

The traveling salesperson problem asks that a Hamiltonian circuit of least total weight be found for a graph. What if you didn't need to visit every vertex in the graph and return back to the starting point, but instead, you needed only to find the shortest path from one vertex in the graph to another? Does an efficient method of solving this type of problem exist? The answer is yes, and the algorithm used in finding the shortest path from a given vertex of a graph to any other vertex in that graph is attributed to E. W. Dijkstra.



E. W. Dijkstra, born in the Netherlands in 1930, is considered one of the original theorists of modern computer science. He first published his shortest path algorithm in a German mathematics journal in 1959, and since that time, he has received many honors and awards. Dijkstra currently serves as professor of mathematics at the University of Texas, Austin.

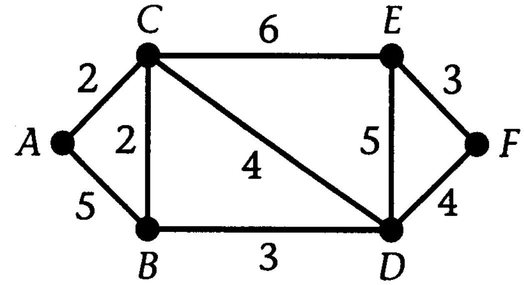
The following algorithm is a modification of Dijkstra's algorithm.

Shortest Path Algorithm

1. Label the starting vertex S and circle it. Examine all edges that have S as an endpoint. Darken the edge with the shortest length and circle the vertex at the other endpoint of the darkened edge.
2. Examine all uncircled vertices that are adjacent to the circled vertices in the graph.
3. Using•only circled vertices and darkened edges between the vertices that are circled, find the lengths of all paths from S to each vertex being examined. Choose the vertex and the edge that yield the shortest path. Circle this vertex and darken this edge. Ties are broken arbitrarily.
4. Repeat steps 2 and 3 until all vertices are circled. The darkened edges of the graph form the shortest routes from S to every other vertex in the graph.

Example

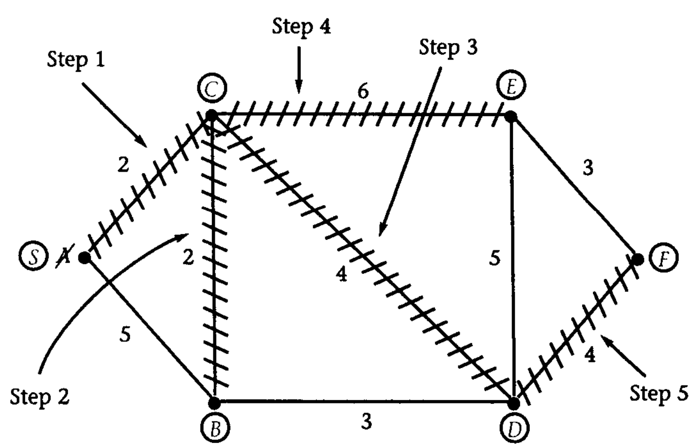
Use the shortest path algorithm to find the shortest path from A to F in the graph.



To find the solution to this problem, begin by circling vertex A and labeling it S. Examine all vertices that are adjacent to S.

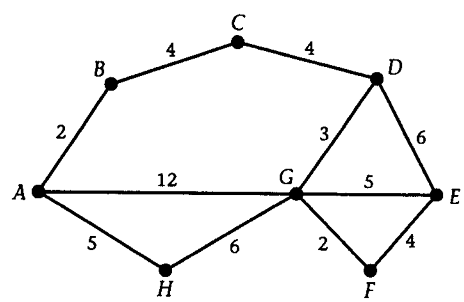
	Adjacent Vertices	Path from *S* to Vertex	Length of Path
Adjacent to *S*	*B*	*SB*	5
	C	*SC*	2
1. Circle *C*, darken edge *SC*.			
Adjacent to *S*	*B*	*SB*	5
Adjacent to *C*	*B*	*SCB*	4
	E	*SCE*	8
	D	*SCD*	6
2. Circle *B*, darken edge *CB*.			
Adjacent to *C*	*E*	*SCE*	8
	D	*SCD*	6
Adjacent to *B*	*D*	*SCBD*	7
3. Circle *D*, darken edge *CD*.			
Adjacent to *C*	*E*	*SCE*	8
Adjacent to *D*	*E*	*SCDE*	11
	F	*SCDF*	10
4. Circle *E*, darken edge *CE*.			
Adjacent to *E*	*F*	*SCEF*	11
Adjacent to *D*	*F*	*SCDF*	10
5. Circle *F*, darken edge *DF*.			

The shortest route from *A* to *F* is *A,C,D,F,* and the length is 10. The darkened edges also show the shortest routes from *A* to the other vertices in the graph.

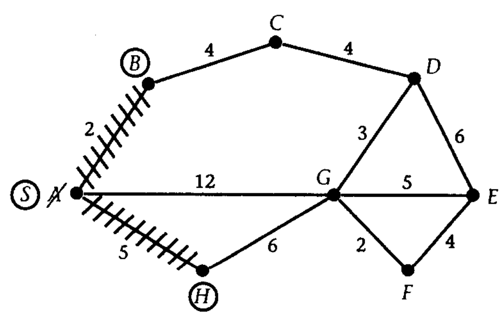


Exercises

1.



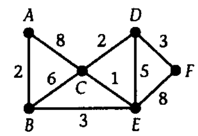
Julian began using the shortest path algorithm to find the shortest route from A to E for the preceding graph. The work that he was able to complete before he had to stop is shown here.



1. $SB - ②$
   $SC - 5$ Circle B, darken SB.
   $SG - 12$
2. $SBC - 6$
   $SG - 12$ Circle H, darken SH.
   $SH - ⑤$
3. $SBC - ?$
   $SG - ?$ Circle?, darken?
   $SHG - ?$

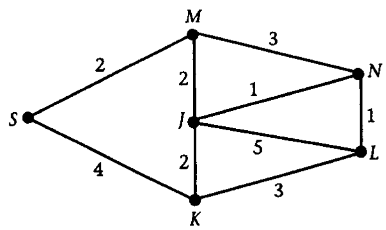
Fill in the missing distances, vertex, and edge in step 3. Then complete Julian's problem of using the shortest path algorithm to find the shortest path from A to E.

2. Use the shortest path algorithm to find the shortest route from A to F.

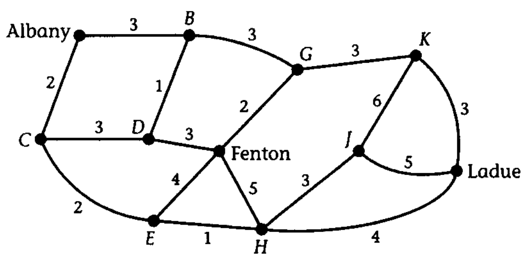


3. When might it not be necessary to repeat the procedure in the algorithm until all of the vertices are circled?

4. Use the shortest path algorithm to determine the shortest distance from S to each of the other vertices in the following graph.



5.

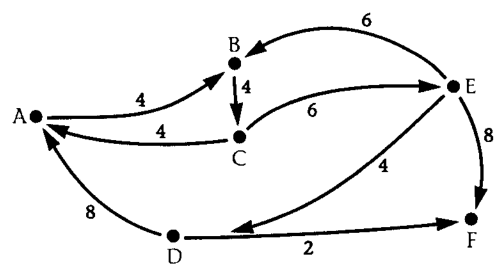


a. Use the shortest path algorithm to find the shortest route from Albany to Ladue in the preceding graph.

b. Assume that it is necessary to travel from Albany to Fenton to deliver a package and then to continue from there to Ladue. Find the shortest route for this trip. Explain why the solution to this question might be different than the shortest route from Albany to Ladue.

6. In the shortest path algorithm, each time you examine the uncircled vertices that are adjacent to the circled ones, you have to recalculate the lengths of the paths from the starting vertex. Explain how the efficiency of the algorithm might be improved by modifying it to avoid such recalculation.

7. The shortest path algorithm can be applied to digraphs if slight modifications are made. Make the appropriate changes, and try your revised algorithm on the following digraph to find the shortest route from *A* to *F*.



8. Mail Packages, Inc., ships from certain cities in the United States to others. A table of the company's shipping costs follows.

		To						
		Albany	Biloxi	Center	Denver	Evert	Fargo	Gale
	Albany	—	7	—	—	4	—	—
	Biloxi	—	—	—	—	—	—	6
	Center	2	—	—	—	2	—	—
From	Denver	—	—	1	—	—	—	—
	Evert	—	—	—	—	—	—	4
	Fargo	—	—	—	—	3	—	2
	Gale	1	6	—	—	—	1	—

Since a package can't be shipped directly from Denver to Biloxi, construct a digraph to represent the cost table and apply the shortest path algorithm to find the least charge for shipping the package.

Projects

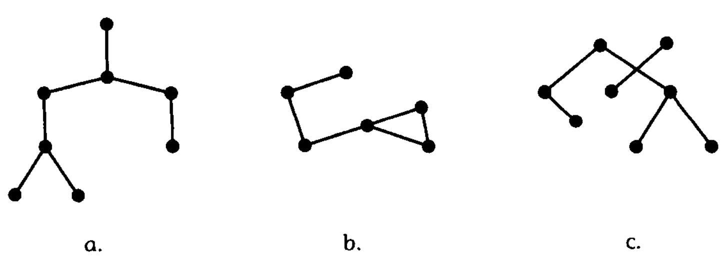
9. Interview several firefighters, ambulance drivers, or paramedics in your community to find out how they determine the shortest route from their facility to an emergency situation. Write a short report on your findings.

In Lesson 5.2, a special type of graph called a tree was used to organize information and list all possible routes for a traveling salesperson problem.

Tree diagrams have been used since ancient times, but it wasn't until the nineteenth century that their properties were studied in detail. In 1847, Gustav Kirchoff used trees in his study of electrical networks. Ten years later, Arthur Cayley used them in his investigation of certain chemical compounds. Today, trees are one of the most useful structures in discrete mathematics and are invaluable to computer scientists. Many computer programmers depend on trees when creating sorting and searching programs.

Before exploring some of the properties and applications of this type of graph, it is necessary to define a tree. Recall that a **cycle** in a graph is any path that begins and ends at the same vertex and no other vertex is repeated. A **tree** is then defined as a connected graph with no cycles.

Which of the following graphs are trees? Why?



Possible trees.

1. Figure 5.9a is a tree because it is a connected graph with no cycles.
2. Figure 5.9b is not a tree because it has a cycle.
3. Figure 5.9c is not a tree because it is not connected.

Trees have many applications in the real world. They can be used to list and count possibilities, as was done in the traveling salesperson problem and as will be done in Chapter 6. They can also be used to model family genealogical histories (Figure 5.10), to structure decision-making processes (Figure 5.11), and to represent chemical compounds (Figure 5.12).

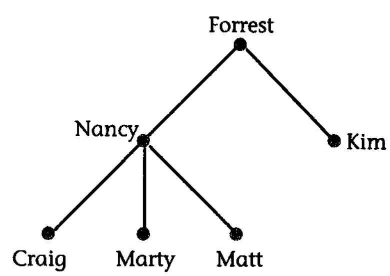


Figure 5.10 Family tree.

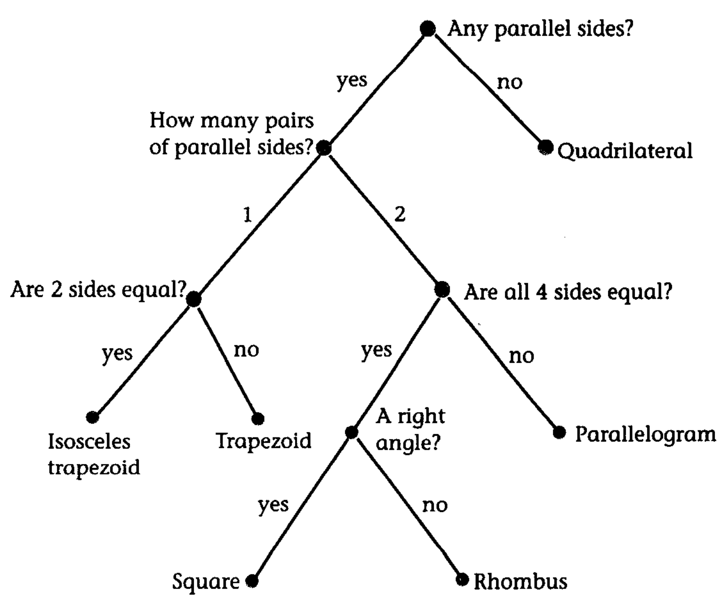


Figure 5.11 Sorting quadrilaterals.

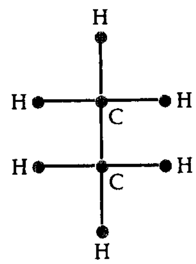
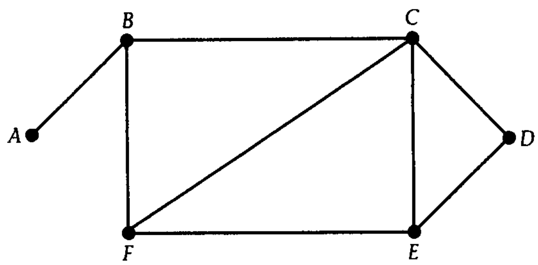


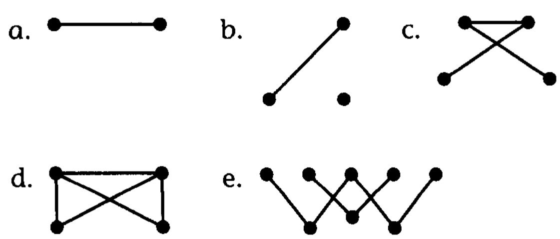
Figure 5.12 C_2H_6 (ethane).

Exercises

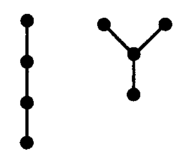
1. Examine the following graph for cycles. List as many as you can find.



2. Determine whether the following graphs are trees. If the figure is not a tree, explain why.



3. There is only one way to draw a tree with two vertices and only one way to draw a tree with three vertices, but there are two distinct trees that can be formed from four vertices. Draw all of the trees that are possible for five vertices. For six vertices.

Number of Vertices	Tree Diagrams
2	
3	
4	

4. Complete the following table for trees with the indicated numbers of vertices.

Number of Vertices	Number of Edges
1	0
2	1
3	
4	
n	

a. How many edges does a tree with 19 vertices have?
b. How many vertices does a tree with 15 edges have?
c. What is the relationship between the number of vertices of a tree and the number of edges?

5. What happens to a tree if an edge is removed from it?

6. Draw a tree with six vertices that has exactly three vertices of degree 1.

7. Complete the following table for trees with the indicated numbers of vertices.

Number of Vertices	Sum of the Degrees of the Vertices	Recurrence Relation
1	0	$S_1 = 0$
2	2	$S_2 = S_1 + 2$
3	4	$S_3 =$ ______
4	______	______
5	______	______
6	______	______

Write a recurrence relation that expresses the relationship between the sum of the degrees of the vertices of a tree with n vertices and the sum of the degrees of the vertices of a tree with $n - 1$ vertices.

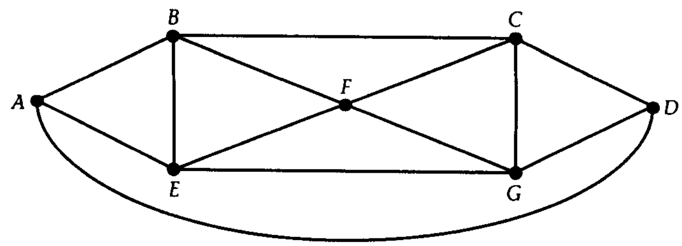
8. Explain why the sum of the degrees of the vertices of a tree with n vertices is equal to twice the number of vertices minus 2. (Hint: Draw several different trees.)

9. Much of the terminology connected with trees is botanical in nature. For instance, a graph that consists of a set of trees is called a **forest,** and a vertex of degree 1 in a tree is called a **leaf**. Draw a forest of three trees. Circle the leaves of your graph.

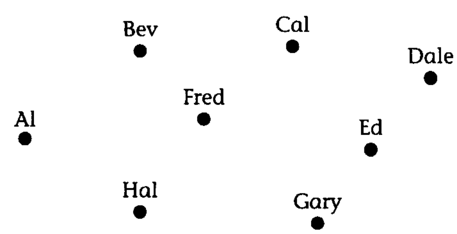
10. In a hierarchical tree as in Figures 5.10 and 5.11 on page 236, it is natural to speak of the **root** of the tree. A tree is rooted when all of the edges are directed away from the chosen vertex (root). In Figures 5.10 and 5.11, the edges are directed downward. Draw a family tree for your family beginning with one of your grandfathers as the root of the tree. What do the leaves of your tree have in common?

11. Refer to the tree in Figure 5.11 on page 236. This graph is called a **decision tree,** and the leaves represent the final outcomes of the different decisions. Using this tree, what is the name of a quadrilateral with two pairs of parallel sides, four sides equal, and no right angles?

12. For the following graph, find two different subgraphs that are trees.



13. Suppose the children in a certain neighborhood want to communicate with one another via their very own communication network. To avoid the expense of connecting each house with every other house, a system needs to be devised that uses as few lines as possible yet allows messages to get to each person. Create such a network for the following houses.



14. Clock solitaire is a card game in which the 52 cards are dealt face down into 13 piles that correspond to the 12 numbers and the center on the face of a clock. In Figure 5.13, clock positions 11 and 12 are identified with the jack and queen, and the thirteenth, or center, pile is identified with the king.

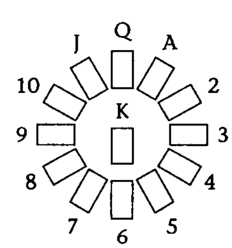


Figure 5.13 Diagram of the 13 piles for the game of clock solitaire.

The game is played by first turning over the top card of the king pile and putting it face up under the pile that corresponds to its value on the clock. Now turn up the top card of the pile under which you just put the card. Continue in this manner. The game is won when you have turned up all 52 cards. If a fourth king is turned up before this happens, play cannot continue, and the game is lost.

In his book *Fundamental Algorithms,* mathematician Donald E. Knuth noted two very interesting things about this game. One is that the probability of winning is 1/13. The other is that by checking the bottom card of the 12 clock piles, you can determine whether you will win the game.

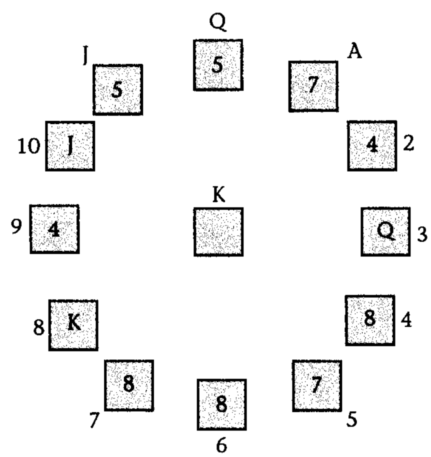


Figure 5.14 One possible configuration of the bottom cards in the game.

To determine whether you can win a game, turn over the bottom card of each pile except for the king pile (see Figure 5.14). Draw an edge from each of the 12 cards to the clock position that corresponds to the card's numeric value. For example, draw an edge from A to 7 (see Figure 5.15). Now redraw the graph with the vertices labeled *A*, 2, 3, . . . *J*, *Q*, and *K*. You will win the game if the resulting graph is a tree that includes all 13 piles (see Figure 5.16).

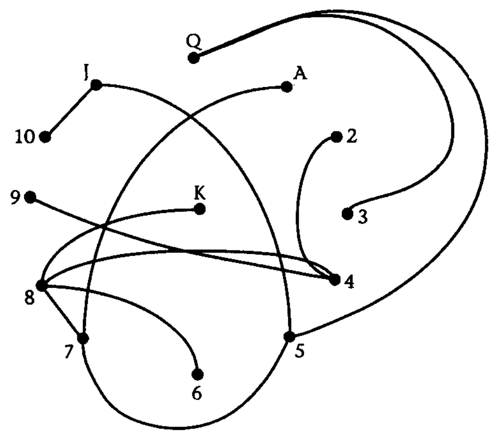


Figure 5.15 Graph showing the edges from the cards to the clock positions.

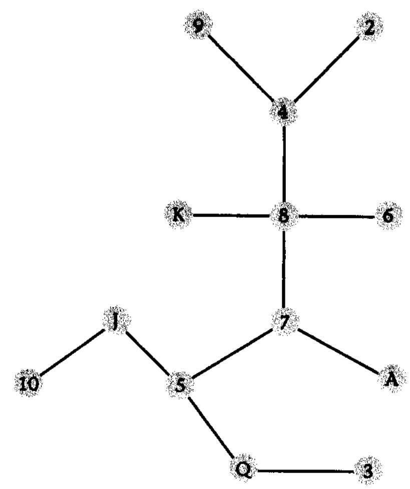


Figure 5.16 Figure showing that the graph in Figure 5.15 is a tree.

Give the game a try, but before you do, record the bottom 12 cards. Predict whether you will win or lose the game by drawing a graph. Notice that only the bottom 12 cards determine your success. The arrangement of the other 40 cards makes no difference.

Projects

15. Research your family tree and report on your discoveries. Be sure to include a graph (tree) in your report to illustrate your findings.

Minimum Spanning Trees

As in many of the previous lessons, this lesson focuses on optimization. Problems and applications here center on two types of problems: finding ways of connecting the vertices of a graph with the least number of edges and finding ways of connecting them with the least number of edges that have the smallest total weight.

Explore This

In making earthquake preparedness plans, the St. Charles County government needs a design for repairing the county roads in case of an emergency. Figure 5.17 is a map of the towns in the county and the existing major roads between them. Devise a plan that repairs the least number of roads but keeps a route open between each pair of towns.

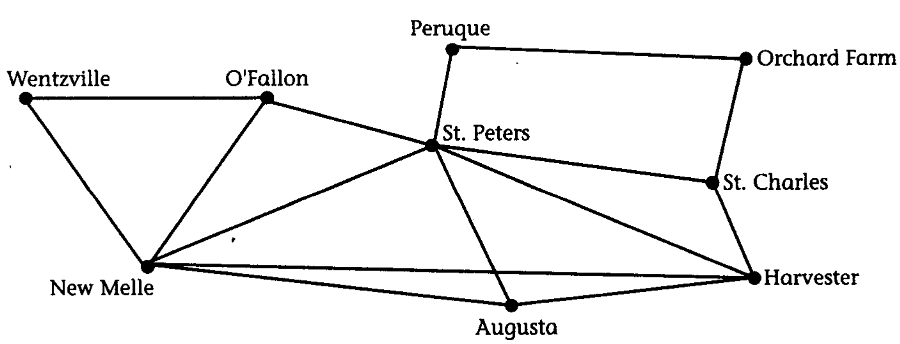


Figure 5.17 The towns in St. Charles County.

Examine your graph. If it connects each of the towns (vertices) and has no cycles, you've found a spanning tree. A **spanning tree** of a connected graph G is a tree that is a subgraph of G and contains every vertex of G. A spanning tree of the graph in Figure 5.17 would represent the minimum number of roads (edges) needed to connect each town in case of an emergency.

Compare your plan with the other plans developed in your class. They should all contain the *same number* of edges but not necessarily the *same* edges. It is possible for a graph to have many different spanning trees. And as you may have guessed, for a graph that is not connected, no spanning tree is possible.

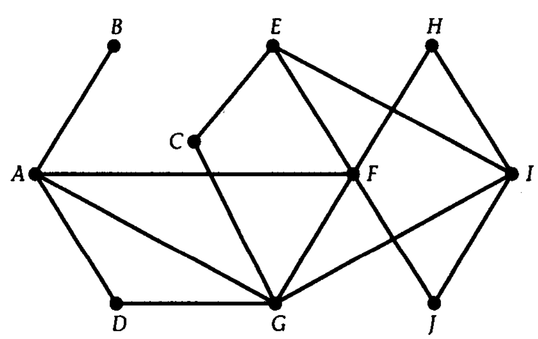
One systematic way to find a spanning tree for a graph is to delete an edge from each cycle in the graph. Unfortunately, this is not an easy procedure for a very large graph. But there are other ways of finding a spanning tree for a graph if one exists. One such method that can be easily adapted to computers is called the *breadth-first search algorithm.*

Breadth-First Search Algorithm for Finding Spanning Trees

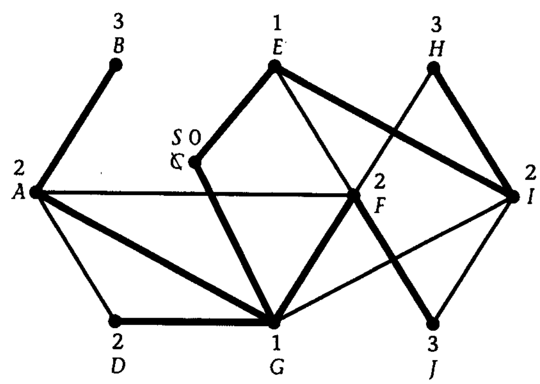
1. Pick a starting vertex, *S*, and label it with a 0.
2. Find all vertices that are adjacent to *S* and label them with a 1.
3. For each vertex labeled with a 1, find an edge that connects it with the vertex labeled 0. Darken those edges.
4. Look for unlabeled vertices adjacent to those with the label 1 and label them 2. For each vertex labeled 2, find an edge that connects it with a vertex labeled 1. Darken that edge. If more than one edge exists, choose one arbitrarily.
5. Continue this process until there are no more unlabeled vertices adjacent to labeled ones. If not all vertices of the graph are labeled, then a spanning tree for the graph does not exist. If all vertices are labeled, the vertices and darkened edges are a spanning tree of the graph.

Example

Use the breadth-first search algorithm to find a spanning tree for the following graph.



As shown in the following figure, the algorithm begins by picking a starting vertex, calling it S, and labeling it with a 0. The labeling and darkening of edges then proceed according to steps 2 to 5 of the algorithm. As you probably noticed, this is not a unique solution. It is just one of the graph's many spanning trees.



Many applications are best modeled with weighted graphs. When this is the case, it is often not sufficient to find just any spanning tree, but to find one with minimal or maximal weight.

Return to the earthquake preparedness situation and reconsider the problem when distances between towns are added to the graph (see Figure 5.18).

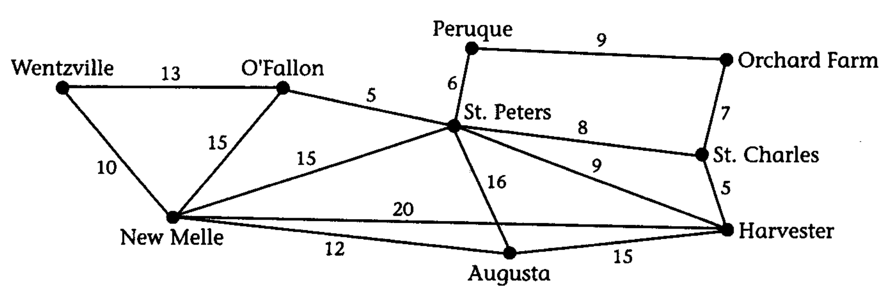


Figure 5.18 Map of St. Charles County with mileage shown.

Refer back to your solution of the original problem and find the total number of miles of road that would need to be repaired if your plan were implemented. Compare your plan with others in your class. Which plan or plans yield the minimum number of miles?

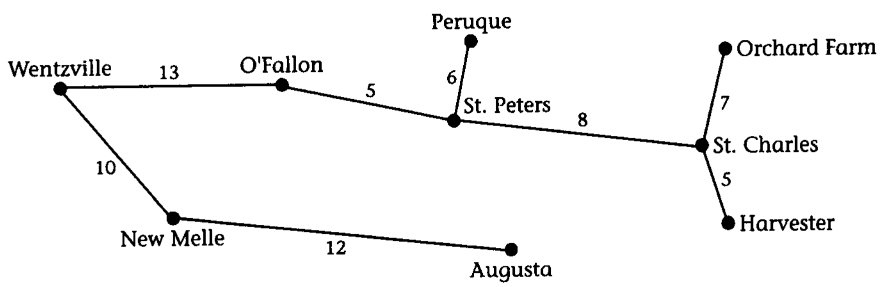


Figure 5.19 Spanning tree of minimum weight for the towns in St. Charles County.

For this particular problem, the minimum possible number of miles of road is 66, and a spanning tree with that total weight is shown in Figure 5.19.

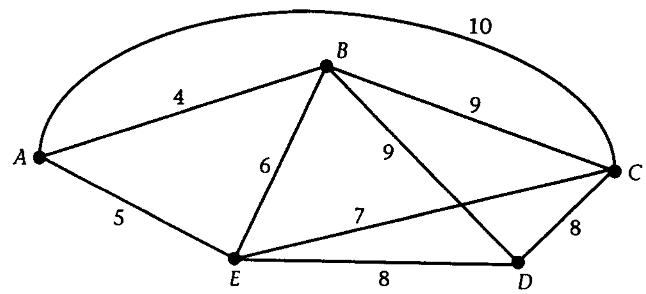
A spanning tree of least, or minimal, weight is called a **minimum spanning tree**. One algorithm for finding a minimum spanning tree for a graph is known as Kruskal's algorithm, named after its designer, Joseph B. Kruskal, a leading mathematician at Bell Laboratories.

Kruskal's Minimum Spanning Tree Algorithm

1. Examine the graph. If it is not connected, there will be no minimum spanning tree.
2. List the edges in order from shortest to longest. Ties are broken arbitrarily.
3. Darken the first edge on the list.
4. Select the next edge on the list. If it does not form a cycle with the darkened edges, darken it.
5. For a graph with n vertices, continue step 4 until $n - 1$ edges of the graph have been darkened. The vertices and the darkened edges are a minimum spanning tree for the graph.

Example

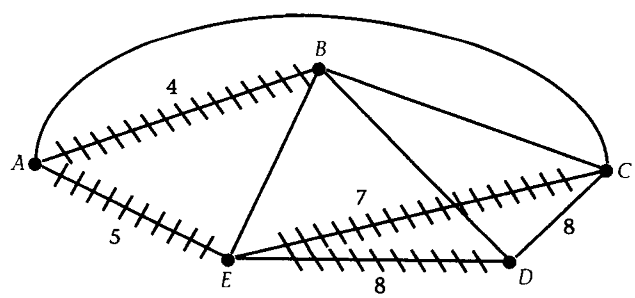
Use Kruskal's algorithm to find a minimum spanning tree for the following graph.



There are five vertices in the graph, so four edges must be chosen. List the edges from shortest to longest. First on the list is *AB* (4). Darken it. Then darken *AE* (5). The next shortest edge is *BE*, but if picked, it will form a cycle. So pick *EC* (7). For the last edge there are two edges of length 8. Either *CD* or *ED* can be darkened. The darkened edges of the following graph form one of the minimum spanning trees of the graph. It has a minimal weight of 4 + 5 + 7 + 8 = 24.

List of Edges from Shortest to Longest

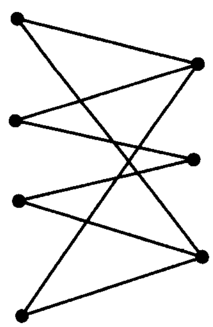
Edge	Length
AB	4
AE	5
BE	6
EC	7
CD	8
ED	8
BD	9
BC	9
AC	10

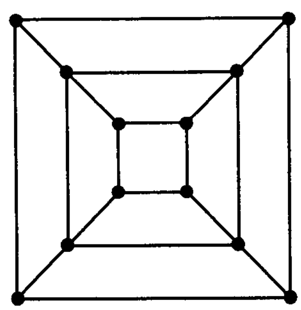


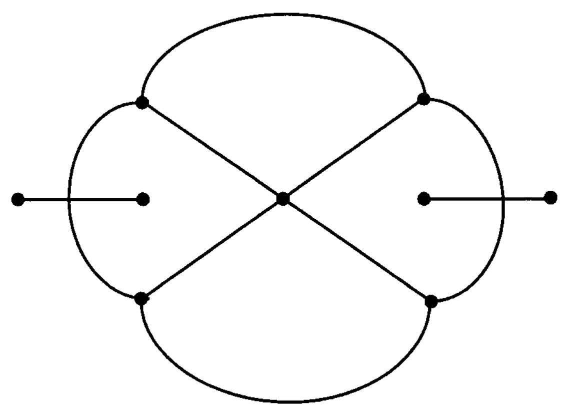
Notice that both Kruskal's and Dijkstra's (Lesson 5.4) algorithms produce spanning trees. But unlike Dijkstra's shortest path algorithm, which gives you a spanning tree of shortest paths, Kruskal's algorithm yields a spanning tree of minimal total weight.

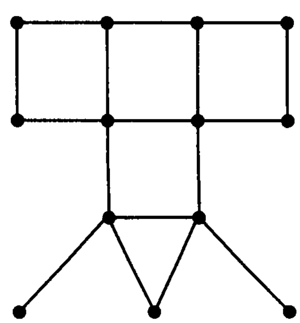
Exercises

In Exercises 1 through 4, find a spanning tree for each graph if one exists.

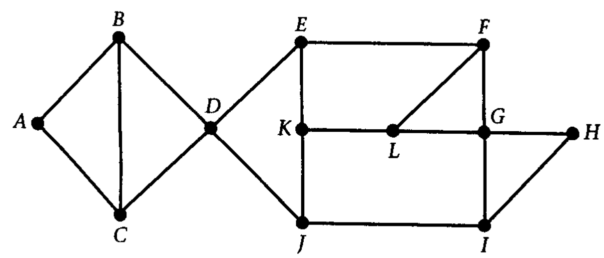
1. 

2. 

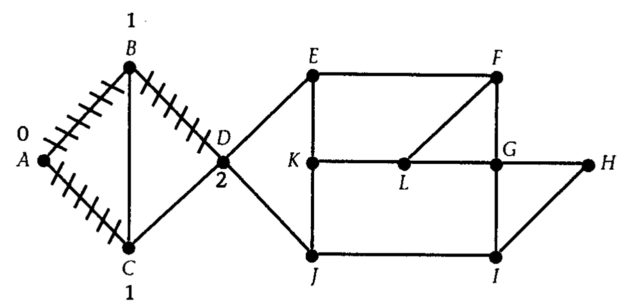
3. 

4. 

5. Draw a spanning tree for a K_4 graph.

6. 

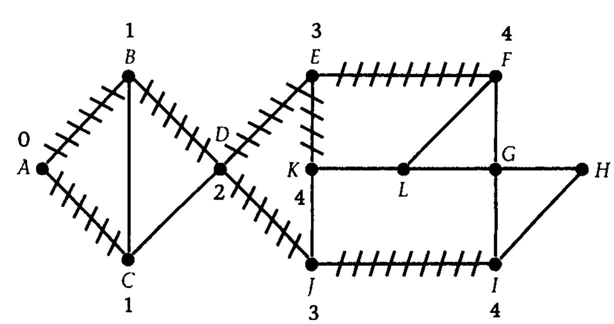
Sid began using the breadth-first search algorithm to try to find a spanning tree for the preceding graph. He began with vertex A, labeled it with a 0, and labeled B and C with 1s. He then darkened edges AB and AC, looked for vertices adjacent to the 1s, and selected vertex D. He labeled it with a 2 and darkened edge BD.



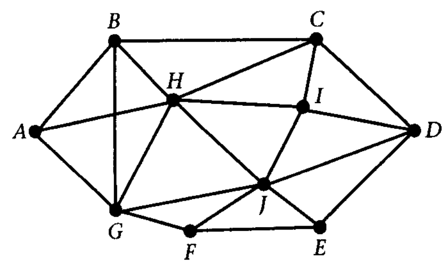
a. Could Sid have darkened CD instead of BD?

Copy Sid's graph and complete the search for Sid by answering the following questions.

b. Which vertices receive 3s for labels? Label these vertices.
c. Which edges subsequently are darkened? Darken these edges.
d. Three vertices should be labeled 4. Which ones? Label these vertices and darken the appropriate edges. Your graph could look like:



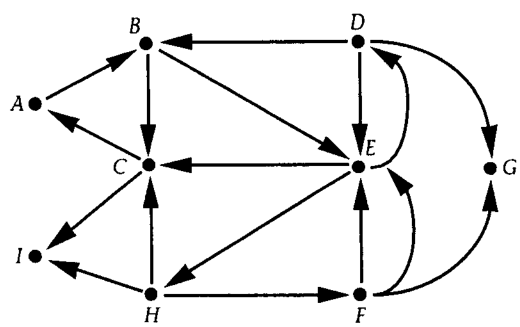
Continue the algorithm until all vertices are labeled. Check your darkened edges to make sure they form a spanning tree.

7. 

Use the breadth-first search algorithm to find a spanning tree for this graph. Begin at vertex *A*.

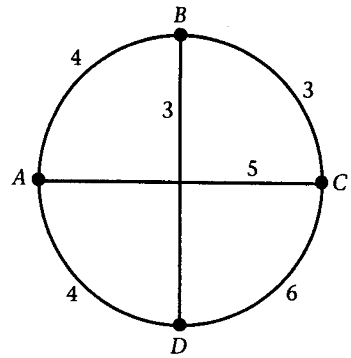
8. Use mathematical induction on the number of edges to prove that every connected graph has a spanning tree.

9. The breadth-first search algorithm can be applied to digraphs if slight changes are made. Modify the algorithm on page 245 so that it can be used with digraphs. Apply your modified breadth-first search algorithm to the following digraph.

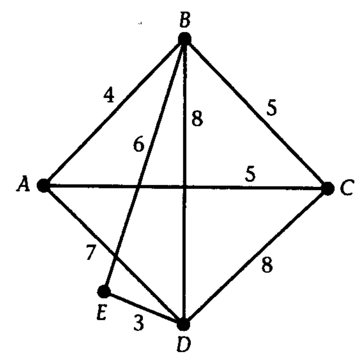


Use Kruskal's algorithm to find a minimum spanning tree for the graphs in Exercises 10 through 12. What is the minimal weight in each case?

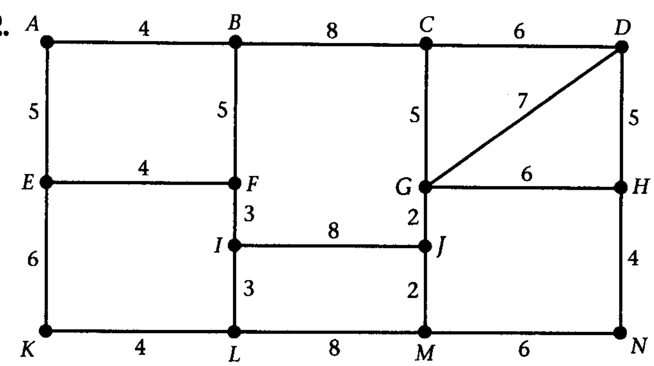
10.



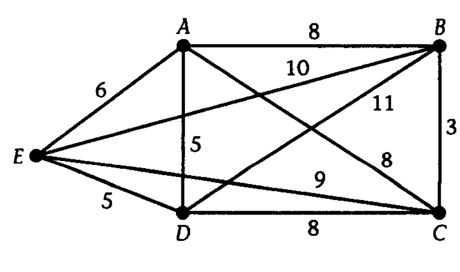
11.



12.



13. The computers in each of the offices at Pattonville High School need to be linked by cable. The following map shows the cost of each link in hundreds of dollars. What is the minimum cost of linking the five offices?



14. Suppose that the cable in Exercise 13 was installed by a disreputable firm that used only the most expensive links. What would be the maximum cost for the four links?

15. How might Kruskal's minimum spanning tree algorithm be modified to make it a maximum spanning tree algorithm?

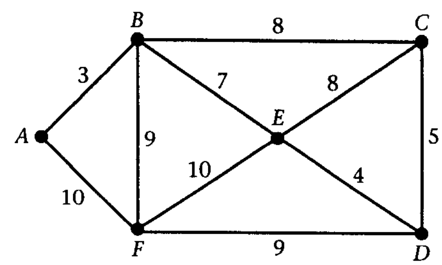
Another algorithm that can be used to find a minimum spanning tree is attributed to R. C. Prim, a mathematician at the Mathematics Center at Bell Labs.

Prim's Minimum Spanning Tree Algorithm

1. Find the shortest edge of the graph. Darken it and circle its two vertices. Ties are broken arbitrarily.
2. Find the shortest remaining undarkened edge having one circled vertex and one uncircled vertex. Darken this edge and circle its uncircled vertex.
3. Repeat step 2 until all vertices are circled.

16. Use Prim's algorithm to find the minimum spanning tree for Exercise 10 and then for Exercise 11.

17. When the shortest path algorithm from Lesson 5.3 is applied until all vertices of a graph are used, it yields a spanning tree of the graph. Is it always a minimum spanning tree?



Check your answer to this question by doing the following.

a. Find a minimum spanning tree of the graph above using either Kruskal's or Prim's algorithm. What is the total weight of the minimum spanning tree?
b. Find the shortest route from A to each of the other vertices using the shortest path algorithm (page 230). Give the lengths of each of these routes.
c. Is the shortest route tree from A to each of the other vertices a minimum spanning tree of this graph? Explain why or why not.

18. Traveling salesperson problems, shortest route problems, and minimum spanning tree problems are often confused because each type of problem can be solved by finding a subgraph that includes all of the vertices of the graph. Compare and contrast what each type of problem asks and when each type of problem is used.

Project

19. In this lesson, you have applied two of the three classical minimum spanning tree algorithms, Kruskal's and Prim's. The third algorithm of this group was designed by O. Borůvka. Investigate Borůvka's algorithm, learn to apply it, and report on how it differs from Kruskal's and Prim's algorithms.

Lesson 5.6

Binary Trees, Expression Trees, and Traversals

Decision trees and family trees are two examples of a special kind of tree known as a **rooted tree**. A rooted tree is a directed tree in which every vertex except the root has an indegree of 1, while the root has an indegree of 0. Figure 5.20 shows an example of a rooted tree in which vertex *R* is the root.

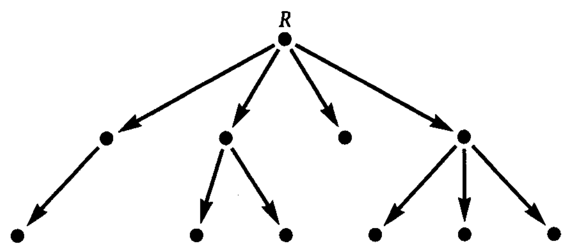


Figure 5.20 Rooted tree.

Since all edges are directed away from the root, it is not necessary to draw the arrowheads on the ends of the edges (see Figure 5.21).

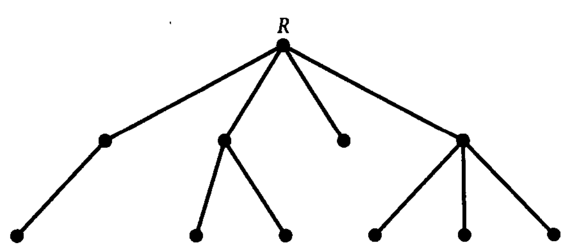
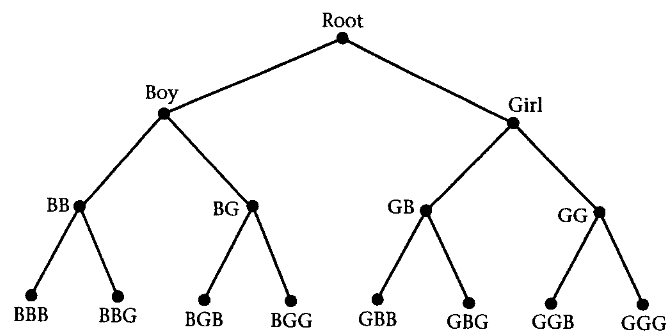


Figure 5.21 Rooted tree without arrowheads.

Rooted trees are used to model situations that are multistaged or hierarchical in structure.

Example

A couple decides to have three children. What are the possible outcomes?

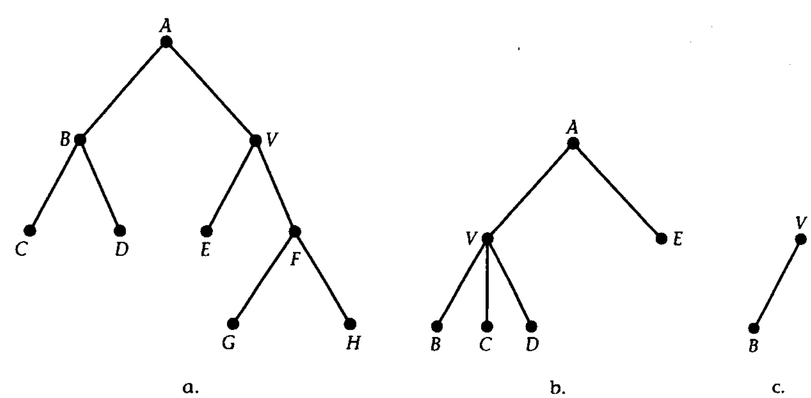


The eight possible outcomes are shown on the vertices of the tree which have outdegrees of 0.

In a rooted tree, a vertex V is said to be at level K if there are K edges on the path from the root to V. The root is at level 0, and the vertices adjacent to the root are at level 1. If a vertex V is at level 4, then any vertex adjacent to V at level 3 is called the **parent** of V, and any adjacent vertex of level 5 is called a **child** of V. A rooted tree in which each vertex has at most two children is called a **binary tree**.

Example

Which trees are binary trees? For those that are binary trees, name the parent of V and the children of V.



All three of these graphs are trees, but only Figures a and c are binary trees. In a, the parent of V is A and the children are E and F. In c, V has no parent and B is the only child.

In computer science applications, binary trees are used to evaluate arithmetic expressions. When you write the expression $(4 + 6) * 8 - 4/2$, you understand how to find its value because you are familiar with the order of operations for expressions. Unfortunately, a computer cannot efficiently imitate your methods. However, if an expression is represented as a binary tree, the computer can quickly and efficiently evaluate it.

To represent the expression $(4 + 6) * 8 - 4/2$ as a binary tree, first find the operation in the expression that is performed last. Make that operation the root of the tree. The right and left sides of this operation become the children of the root (see Figure 5.22).

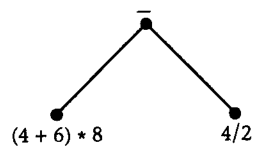


Figure 5.22 First step of representing an expression as a binary tree.

Continue this recursive process of placing operations at each internal vertex (the children) and putting operands on the leaves until no expression that contains operations appears on the leaves (see Figure 5.23).

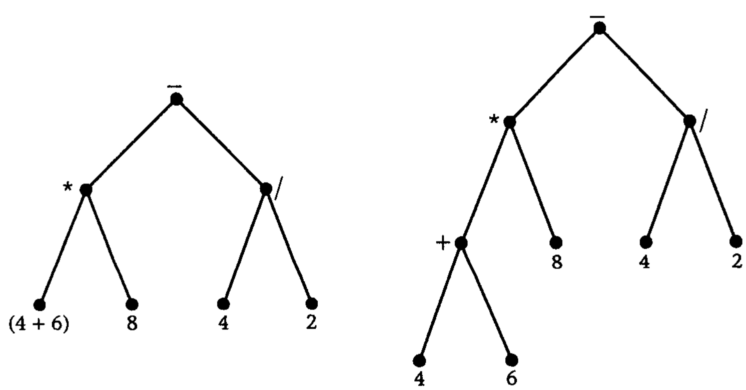


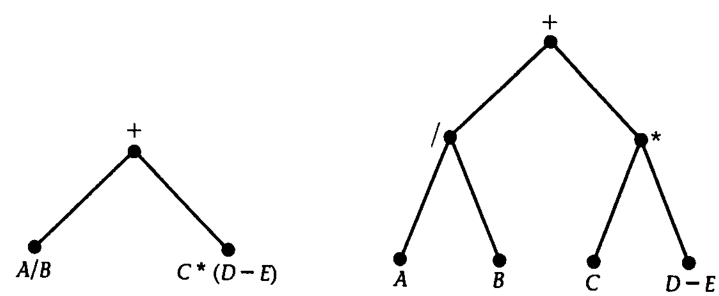
Figure 5.23 Continuing the recursive process.

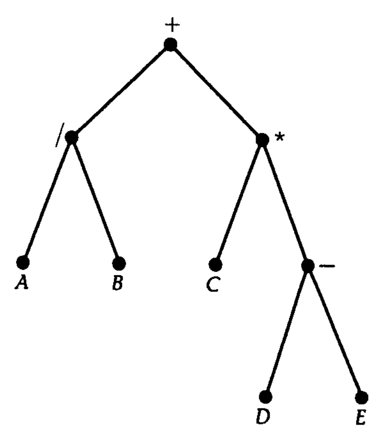
The final binary tree in Figure 5.23 is called an **expression tree**.

Example

Represent $A/B + C * (D - E)$ as an expression tree.

The solution is:





Tree Traversals

Once an expression is represented by a binary tree, the computer must have some systematic way of "looking at" the tree in order to find the value of the original expression. This organized procedure for obtaining information by visiting each vertex of the tree exactly once is called a **traversal** of the graph.

There are many different types of traversals, including one called a **postorder traversal**. This traversal differs from other traversals in that it visits the left child of the tree first, then the right child, and finally the parent or root (see Figure 5.24).

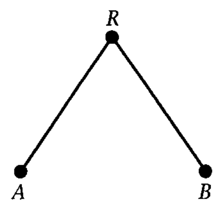


Figure 5.24 Postorder traversal *A, B, R.*

To find the postorder listing of the vertices of the tree in Figure 5.25, begin by moving to the left subtree of *A* and doing a postorder traversal on that subtree, which has *B* as its root. This requires you to branch to the left subtree of *B* and do a postorder traversal. Since the left subtree of *B* consists of only the vertex *D*, visit that vertex by numbering it with a 1. Now go to the right subtree of *B* and do a postorder traversal. Again, this subtree consists of only one vertex. Visit *E* and number it with a 2. Since the left and right children of *B* have been traversed, visit *B* (the root of that subtree). Number it 3.

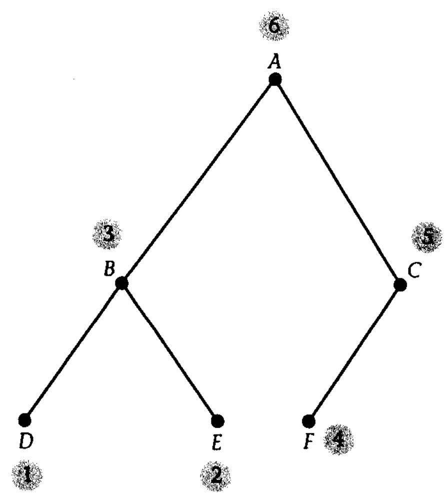
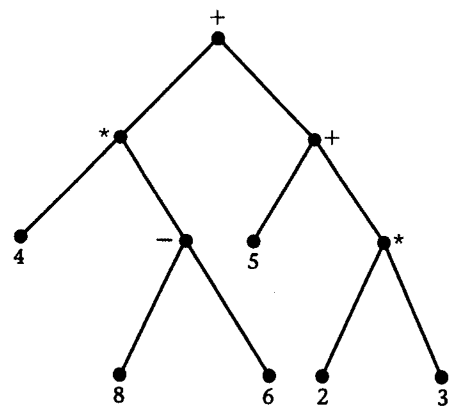


Figure 5.25 Postorder traversal of a binary tree.

You have traversed the left subtree of *A* and must now traverse the right subtree. To do this, you begin with a postorder traversal on the subtree which has *C* as its root. Move to the left subtree of *C*, visit *F*, and number it 4. Since there is no right subtree of *C*, visit the root *C*, and number it 5. Since both the left and right subtrees of root *A* have been visited, *A* can now be visited and numbered with a 6. The postorder traversal is complete and the postorder listing is *DEBFCA*.

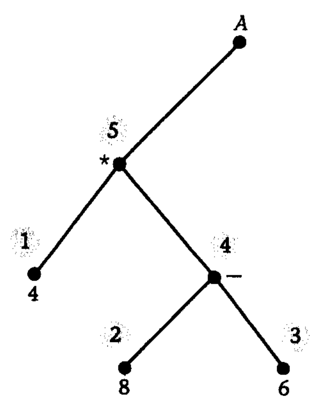
Example

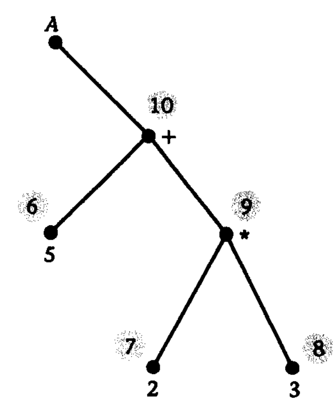
Give a postorder listing for the following expression tree.



The following is the solution.

Left subtree of *A* | Right subtree of *A*





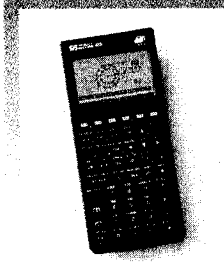
The root is visited last and 11 is assigned to *A*. The postorder listing is 4 8 6 − ∗ 5 2 3 ∗ + +.

The notation obtained by doing a postorder traversal is known as **reverse Polish notation (RPN)**. The notation was so named because it was introduced by the Polish mathematician Jan Lukasiewiez. This notation with operations next to each other and no parentheses may look strange to you, but to owners of certain calculators, this notation is familiar and easy to use. RPN works well with calculators and computers because no parentheses are ever needed to indicate the desired order of operations.

So how do you find the value of the expression 4 8 6 − ∗ 5 2 3 ∗ + +? To evaluate RPN, scan the expression from the left until you find two numbers followed by an operation sign, in this case, 8 6 −. This says to you to take the 8 and 6 and subtract. Substitute the result, 2, back in the expression and repeat the process. This continues until you have evaluated the expression.

4 (8 6 −) * 5 2 3 * + +
(4 2 *) 5 2 3 * + +
8 5 (2 3 *) + +
8 (5 6 +) +
(8 11 +)
19

Point of Interest



Some calculators designed by Hewlett Packard, such as the HP 48G series of graphic calculators, support RPN as their primary entry system.

People who become accustomed to using this type of notation find it very quick and convenient to use because there is never a question about the order in which to perform the operations.

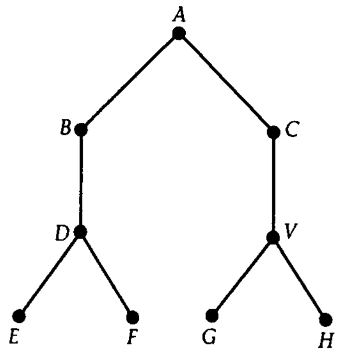
Exercises

1. Tony wants to buy a car. He has the options of two different brands of radios and four different exterior colors. Draw a tree diagram to show all possible outcomes of choosing a radio and a color for the car.

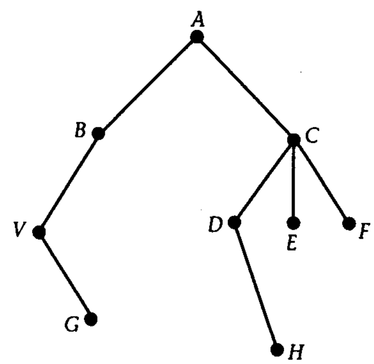
2. A coin is tossed three times. Draw a tree diagram to show the possible outcomes.

In Exercises 3 through 6, examine each tree. If the tree is a binary tree, (a) give the level of vertex V, (b) name the parent of V, and (c) name the children of V.

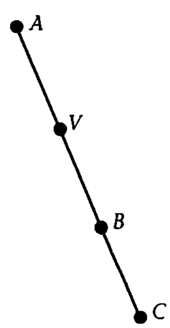
3.



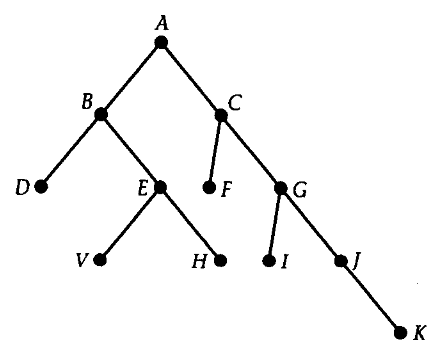
4.



5.



6.



7. Jeff's brother Tom has a first-grade spelling book that contains five chapters. Each odd-numbered chapter has two lessons and each even-numbered chapter has three lessons. The second lesson of each chapter has two questions whereas all others have one. Draw a rooted tree that models Tom's book. How many questions are in the book?

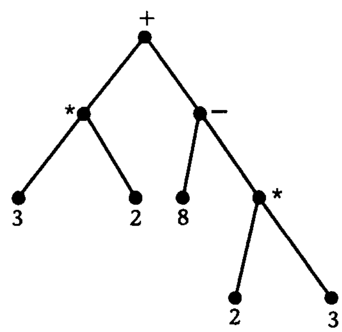
In Exercises 8 through 11, represent the expression as a binary expression tree.

8. $(2 - 5) * (4 + 7)$ **9.** $(2 + 3) * 4$

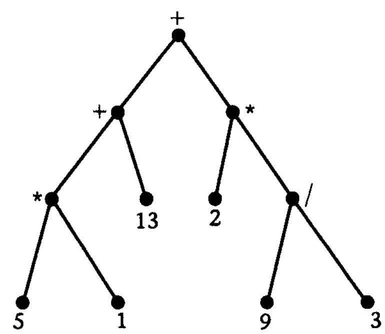
10. $2 + 3 * 4 - 6/2$ **11.** $A * B + (C - D/E)$

In Exercises 12 through 14, find the postorder listings for each binary tree.

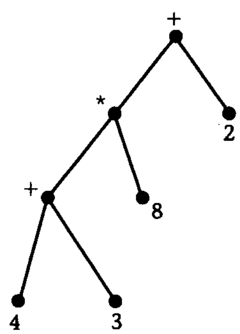
12.



13.



14.



15. Evaluate the following reverse Polish notations.

a. 6 2 − 7 ∗ 3 2 + + b. 6 5 4 3 2 − + / +

c. 1 2 + 4 3 − + 6 2 / 2 + + d. 4 3 + 8 2 − + 4 + 3 −

16. Give the reverse Polish notation for each of the following expressions.

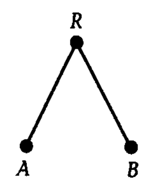
a. 2 + 3 ∗ 6 − (4 + 1) b. (5 − 3) ∗ 2 + (7 − 6/2)

17. Construct an expression tree that would have reverse Polish notation

$$A\ B * C\ D + E - +$$

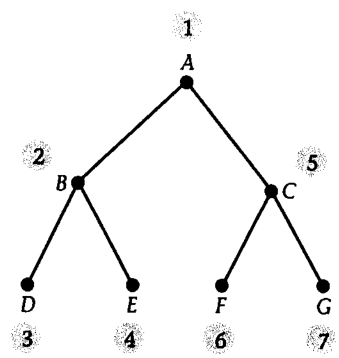
18. Construct a binary tree that has *ABC* as its postorder listing. Is your answer unique? If not, construct an additional tree(s).

19. A traversal that visits first the parent or root of the tree, then the left child, and finally the right child is called a **preorder traversal**.

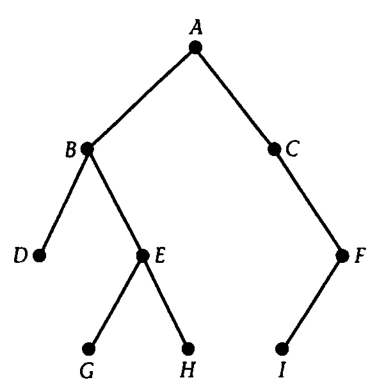


Preorder traversal
R, A, B

The preorder listing for the following binary tree is *ABDECFG*.



Find the preorder listing for the following binary tree.



20. Find the preorder listings for the binary trees in Exercises 12 through 14.

21. The notation obtained from a preorder traversal is called **Polish notation**. To evaluate the expression, scan it from the left until you come to an operation followed by two numbers. Perform that operation, place the result back in the expression and continue. For example, * 2 3 is evaluated as 6. Complete the following evaluation.

+ + 4 (* 2 3) + 5 / 6 3
+ (+ 4 6) + 5 / 6 3
+ 10 + 5 / 6 3

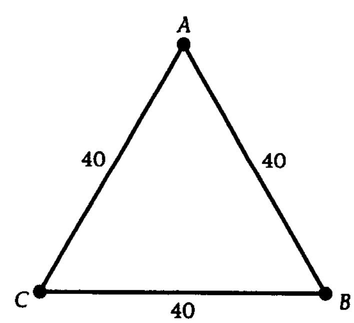
22. Evaluate the following Polish notations.
a. + * 3 2 − 8 * 2 3
b. + / 6 3 + 4 3

Project

23. A postorder traversal of an expression tree yields reverse Polish notation, in which the operations follow the operands. A preorder traversal yields Polish notation, in which the operations precede the operands. Create a traversal rule that yields a notation in which the operations are between the operands. In a report, describe your procedure and show how it works on an expression tree. Also find a rule that evaluates your listings. Try your rule on several different expressions. Does it work for all of them? If not, explain why you think it's flawed.

Steiner Trees

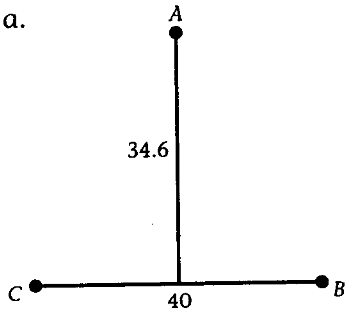
Dr. Terry has three computers in her office which need to be networked. If the following graph shows the shortest distance between each computer, what is the minimum amount of cable required?



Finding a minimum spanning tree to link the three computers yields a total length of 80 feet of cable, but is there a better way?

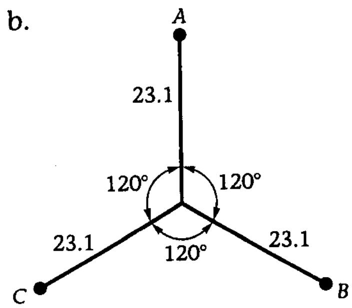
The answer to this question is yes, if you do not have to follow the edges in the graph, and creating a junction someplace other than at one of the computers is not a problem. For example, if the cables for these computers can be placed and joined anywhere in the room, you can use less than the 80 feet required for the minimum spanning tree solution (see Figures a and b).

a.



Total length of cables: 74.6 ft.

b.



Total length of cables: 69.3 ft.

The tree using the newly created junction point, such as the one shown in Figure b, is known as a **Steiner tree,** and the junction point where the three edges meet at 120° angles is called a **Steiner point**.

A Steiner point inside a triangle can be found using the following procedure, known as the Torricelli procedure.

> Assume that the largest angle of triangle *ABC* is less than 120° and that *BC* is the longest side of the triangle. On the longest side of the triangle *BC* construct an equilateral triangle *BCP*. Circumscribe a circle around triangle *BCP*. Join *P* to *A*. The point where the line segment *PA* intersects the circle is the desired Steiner point.

Creating Steiner trees brings up many questions that can be explored using a drawing utility such as the Geometer's Sketchpad or Cabri:

- If the computers are positioned on the vertices of a triangle that is not equilateral, will there be a Steiner tree?
- In some three-point cases, the minimum spanning tree solution is the best networking solution. When will this happen?
- If there were four computers to network instead of three, how many Steiner points would there be?
- Will there ever be more than one Steiner tree? If so, will they always be equal in length?

Of course, the questions continue to expand as the number of vertices increases. As you may have already discovered, the optimal way of connecting the vertices of a graph (or computers, in this case) is either to find the minimum spanning tree if no interior junctions are allowed or to construct an optimal Steiner tree.

As the number of vertices in a graph increase, the number of Steiner trees increases very rapidly. For example, the number of Steiner trees in a graph with ten vertices totals in the millions. Therefore, finding the Steiner tree with the least weight becomes very complicated. Since their discovery, mathematicians have made significant findings about Steiner trees and have even created efficient algorithms that approximate optimal solutions. But unfortunately, as with the traveling salesperson problem, there is no efficient algorithm that always finds the desired minimum distance.

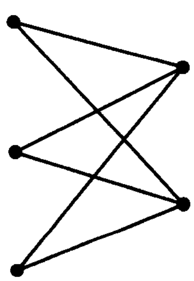
Chapter 5 Review

1. Write a summary of what you think are the important points of this chapter.

2. Show that the graph described by the following adjacency matrix is planar.

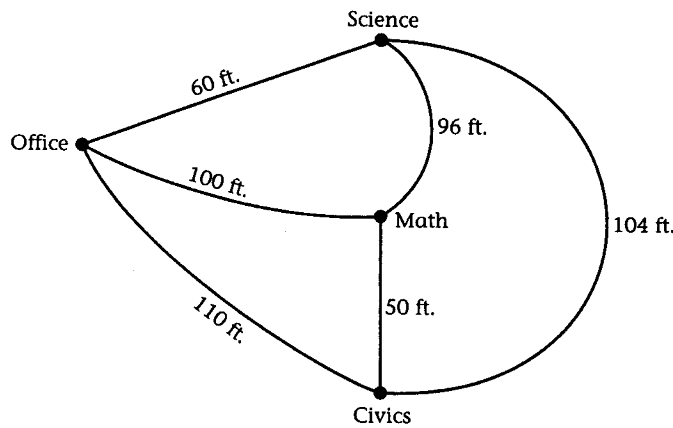
$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

3. What is the chromatic number for a tree with five vertices? Any odd number of vertices? Any even number of vertices? Any tree with two or more vertices?

4. 

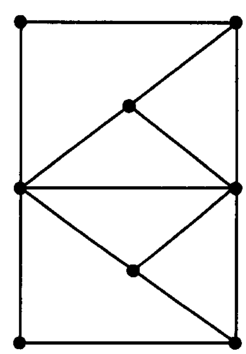
   a. Explain why this graph is a bipartite graph.
   b. Is this graph a complete bipartite graph? Explain why or why not.
   c. Is this graph planar? If so, find a planar drawing for the graph.
   d. What is the chromatic number for this graph?

5. Mr. Gonzalez, the principal at Central High School, leaves his office once an hour to visit the math, science, and social studies classrooms, and then returns to his office. The distances between rooms are shown on the following graph.



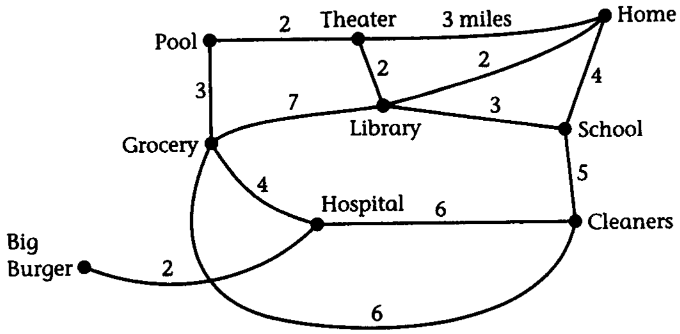
a. Find the shortest route possible for Mr. Gonzalez.
b. What is the total distance of the shortest route in part a?
c. What kind of circuit does Mr. Gonzalez make?

6. Find a spanning tree for the following graph if one exists.



7. How many different spanning trees are there for a cycle with 3 vertices? With 4 vertices? With 5 vertices? With n vertices?

8. When given the position where it is currently located and its destination, a certain robot car is programmed to find the shortest path for the trip. The routes that the car can travel are shown on the following graph.

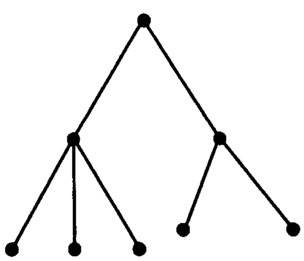


a. Use inspection to find the shortest path from Home to Big Burger.
b. What is the minimum distance from Home to Big Burger?
c. Use the shortest path algorithm (page 230) to find the shortest paths from Home to each of the other locations on the graph.

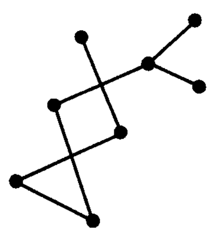
9. Assume that all locations represented by the graph in Exercise 8 need to be connected by cable. Find the minimum amount of cable needed to link the nine locations.

10. Are the following graphs trees? Explain why or why not.

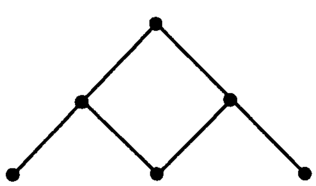
a.



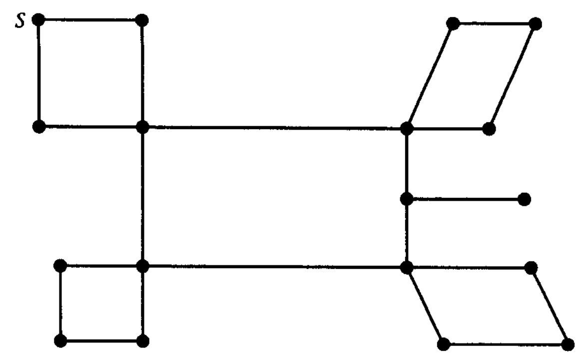
b.



c.

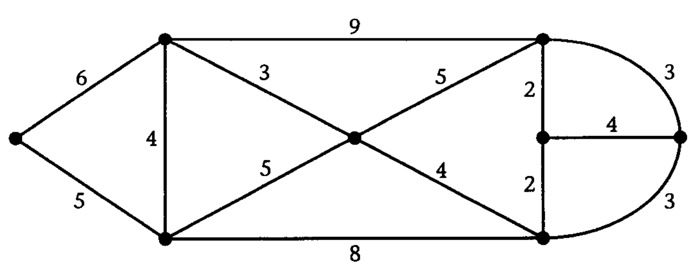


11. Use the breadth-first search algorithm from page 245 to find a spanning tree for the following graph. Begin the algorithm at the vertex labeled *S*.



12. Draw a tree with eight vertices that has exactly four vertices of degree 1.

13. The vertices of the following graph represent buildings on a small college campus. Administrators at the campus want to connect the buildings with fiber-optic cable and are interested in finding the least expensive way of doing so. The costs of connecting buildings (in thousands of dollars) are shown as weighted edges of the graph.



a. Use one of the spanning tree algorithms to find a minimum spanning tree for the graph.
b. What is the total cost of connecting the buildings?

14. Create a problem that can be solved by using one of the minimum spanning tree algorithms and find the solution to your problem. Then give your problem to a classmate and ask him or her to solve it. If the answer differs from yours, determine the correct solution.

15. Represent (4 − 3) ∗ 8 + 5 as a binary expression tree.

16. Evaluate the reverse Polish notation

$$7\ 1 + 3 * 2\ 4 + -$$

17. Create a mathematical expression of your own, represent it with an expression tree, and find the postorder listing for the tree. Give your listing to another student in your class and have him or her evaluate the reverse Polish notation. Check the value of the notation with the value of your original mathematical expression.

Bibliography

Chartrand, Gary. 1977. *Introductory Graph Theory*. New York: Dover.

Chavey, Darrah. 1992. *Drawing Pictures with One Line*. (HistoMAP Module #21). Lexington, MA: COMAP.

COMAP. 1997. *For All Practical Purposes: Introduction to Contemporary Mathematics*. 4th ed. New York: W. H. Freeman.

Copes, W., C. Sloyer, R. Stark, and W. Sacco. 1987. *Graph Theory: Euler's Rich Legacy*. Providence, RI: Janson.

Cozzens, Margaret B., and R. Porter. 1987. *Mathematics and Its Applications*. Lexington, MA: COMAP.

Cozzens, Margaret B., and R. Porter. 1987. *Problem Solving Using Graphs*. Lexington, MA: COMAP.

Crisler, Nancy, and Walter Meyer. 1993. *Shortest Paths*. Lexington, MA: COMAP.

Dossey, John, A. Otto, L. Spence, and C. Vanden Eynden. 1996. *Discrete Mathematics*. 3rd ed. New York: Harper Collins.

Francis, Richard L. 1989. *The Mathematician's Coloring Book*. (HiMAP Module #10). Lexington, MA: COMAP.

Ore, O. 1990. *Graphs and Their Uses*. Washington, DC: Mathematical Association of America.

Tannenbaum, Peter, and Robert Arnold. 1992. *Excursions in Modern Mathematics*. Englewood Cliffs, NJ: Prentice-Hall.