Chapter 2
Business Efficiency

Chapter Outline

Chapter Summary

A *Hamiltonian circuit* in a graph is a simple circuit that contains every vertex of the graph. Unlike the situation with Euler circuits, there are no known conditions that guarantee that a graph has a Hamiltonian circuit. However, it is known that certain graphs have Hamiltonians (e.g., the complete graphs) and that others do not (e.g., the graph displayed in Figure. 2.2(a) in the text).

The *traveling salesman problem* (*TSP*) is to find in a given weighted graph a Hamiltonian circuit of least total weight. The problem is a generalization of the problem of finding the cheapest route for a salesman who must visit clients in several cities and then return home.

A naive solution to the problem is the brute force method (also known as exhaustive search). This method is computationally infeasible, even for relatively small values of n (e.g., $n = 20$). Unfortunately, there is no known algorithm that will generate optimal solutions more quickly. In fact, many experts believe that no such algorithm will ever be found.

Consequently, we use *heuristic algorithms* to solve this problem. Heuristics are fast algorithms but are not guaranteed to produce optimal solutions. Two such algorithms for the TSP are the *nearest-neighbor algorithm* and the *sorted-edges algorithm*. Both of these algorithms are greedy in the sense that each time a choice is made, they make the choice that seems best based on the objective of the problem. Unfortunately, these local best choices do not necessarily combine to give an optimal solution to the TSP.

A *tree* is a connected graph with no circuits. A *spanning tree* in a given graph is a tree built using all the vertices of the graph and just enough of its edges to obtain a tree. The *minimum-cost spanning tree* problem is to find a spanning tree of least total edge weight in a given weighted graph. A sorted-edges greedy approach can be used to get a solution to this problem. It is interesting that this algorithm, developed by Joseph Kruskal, always produces an optimal solution to this problem.

The final topic in this chapter is a lead-in to the next chapter. In a job composed of several tasks (e.g., assembling a bicycle), there is often an order in which the tasks must be performed. This ordering of tasks can be represented by using a *digraph* (short for directed graph). The vertices of the graph represent the tasks, and the edges are directed from one vertex to another. Directed edges are like one-way streets, represented by arrows pointing in the allowable direction of travel. A certain directed path in this graph, the *critical path*, corresponds to the sequence of tasks that will take the longest time to complete. Since our job is not complete until every possible sequence of tasks has been finished, the "length" of the critical path tells us the least amount of time it will take us to complete our job. It is possible for a digraph to have more than one critical path.

Skill Objectives

1. Give the definition of a Hamiltonian circuit.

2. Explain the difference between an Euler circuit and a Hamiltonian circuit.

3 Identify a given application as being an Euler circuit problem or a Hamiltonian circuit problem.

4. Calculate $n!$ for a given value of n.

5. Apply the formula $\frac{(n-1)!}{2}$ to calculate the number of distinct Hamiltonian circuits in a complete graph with a given number of vertices.

6. Define the term *algorithm*.

7. Explain the term *heuristic algorithm* and list both an advantage and a disadvantage.

8. Discuss the difficulties inherent in the application of the brute force method for finding the minimum-cost Hamiltonian circuit.

9. Describe the steps in the nearest-neighbor algorithm.

10. Find an approximate solution to the traveling salesman problem by applying the nearest-neighbor algorithm.

11. Describe the steps in the sorted-edges algorithm.

12. Find an approximate solution to the traveling salesman problem by applying the sorted-edges algorithm.

13. Give the definition of a tree.

14. Given a graph with edge weights, determine a minimum-cost spanning tree.

15. Identify the critical path in an order-requirement digraph.

16. Find the earliest possible completion time for a collection of tasks by finding the critical path in an order-requirement digraph.

17. Explain the difference between a graph and a directed graph.

Teaching Tips

1. Perhaps the most important concept in this chapter (as well as in Chapters 3 and 4) is the notion of an algorithm. Stress that in most large-scale problems, algorithms are implemented on computers. Hence, detailed, step-by-step instructions must be provided, and the computer, having no judgment of its own, is incapable of determining cases in which it might be beneficial to deviate from these instructions.

2. It may be helpful to point out that the graph in Figure 2.3 is neither drawn to scale, nor is it geographically accurate in terms of the positioning of the cities. Because a mathematical graph is a symbolic model, only the fact that there are four vertices in distinct locations needs to be constant. The positioning in this diagram makes the interpretation clear and easy to read.

3. When traveling by air, shortest distance doesn't necessarily correspond with least cost, as is normally the case with automobile travel. As a special project, students can check with an airline about fares between the cities demonstrated in Examples 1 and 3 in the text. Plan a least-cost version of the nearest-neighbor algorithm.

4. It might be helpful to define precisely what is meant by a heuristic algorithm, since it is likely that students have never heard this term before.

5. Note that the nearest-neighbor algorithm starts at a specified vertex and that the route obtained may be different if the starting vertex is changed.

6. When discussing the sorted-edges algorithm, it may be helpful to indicate that a starting point is not critical. After the edges are linked, then the starting point may be selected and the route followed along the Hamiltonian circuit. You might consider discussing the minimum-cost spanning tree concept along with the sorted-edges algorithm to reinforce the concept. A discussion of the logic behind the two conditions below may help students who are having difficulty.

a. Three edges cannot meet at a vertex; if this happened, it would mean that the city in question has been visited more than once. Two edges meeting at a vertex merely provide a way into the city and another way out of the city, whereas the third edge would then lead back to the same city, thus violating the premise of the Hamiltonian circuit.

b. A circular tour cannot be created without all the cities; the creation of a circuit would end the tour, but the tour isn't over until all the vertices have been visited exactly once.

7. The concluding section on critical path analysis is a nice lead-in for Chapter 3; however, if you're running short of time, it can be delayed until then. One advantage of discussing it in this chapter is that students have an opportunity to explore the idea before coupling it with the scheduling concept in Chapter 3.

Research Paper

Have students research another method of finding an efficient way of traversing a graph, such as Prim's method or Dijkstra's algorithm. Students should discuss this method (discovered and rediscovered) and its similarities and differences to those described in the text. Students should also research the lives of the people for whom they are named (Robert C. Prim and Edsger W. Dijkstra).

Collaborative Learning

Hamiltonian Circuits

Begin the lesson by defining Hamiltonian circuits. Either draw the following Hamiltonian circuit diagrams on the board or duplicate the page and distribute it to your students. Working in groups, ask them to find Hamiltonian circuits, if possible. After they decide which graphs have Hamiltonian circuits, ask them if they can find criteria for the existence of such a circuit, similar to those for an Euler circuit. (Of course, no such criteria are known to exist.) The difference between the two problems appears to be minimal, with the focus merely changed from edges to vertices. Many students find it striking that the Euler problem is easily solvable, while the Hamiltonian one is not.

Traveling Salesman Problem

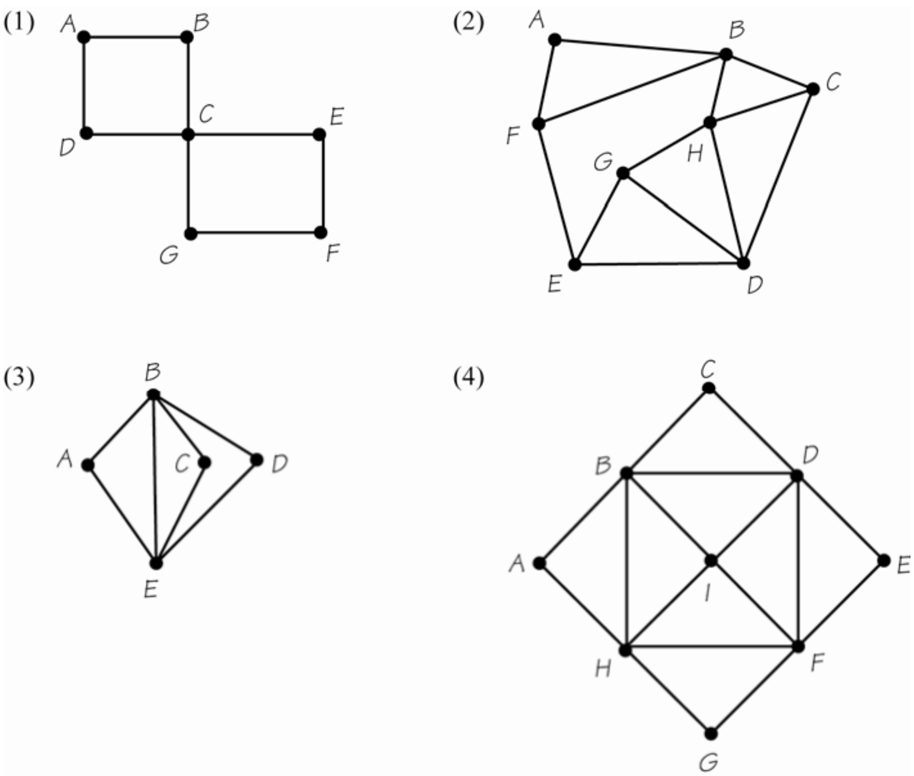
To introduce the traveling salesman problem, ask the students to try to find the minimal Hamiltonian circuit in each of the complete graphs in the following TSP exercise.

Minimum-Cost Spanning Trees

After defining the notion of a minimum-cost spanning tree, but before introducing Kruskal's algorithm, have the students attempt to find the minimum-cost spanning tree for each of the graphs on the next page. Perhaps with some hints they can discover the algorithm themselves.

Hamiltonian Circuits

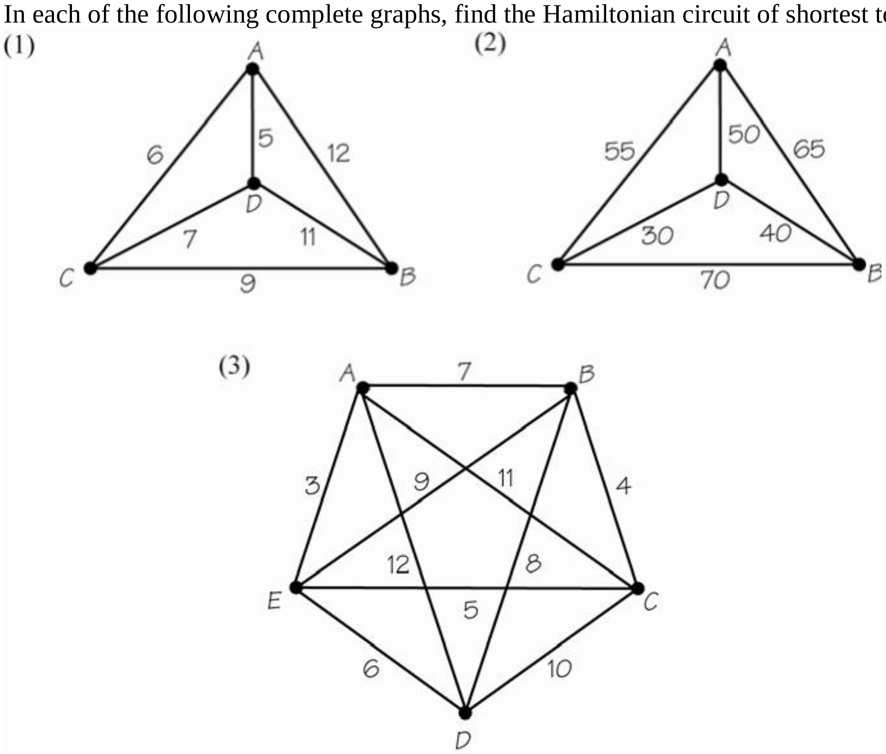
1. Find the Hamiltonian circuits in the following graphs, when possible.



2. Use the campus map as in the exercise from Chapter 1, but now in the context of Hamiltonian circuits. If you wish, you can include distances and turn this into a traveling salesman problem.

Traveling Salesman Problem

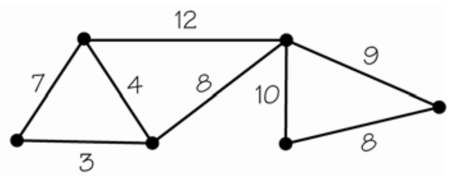
In each of the following complete graphs, find the Hamiltonian circuit of shortest total length.



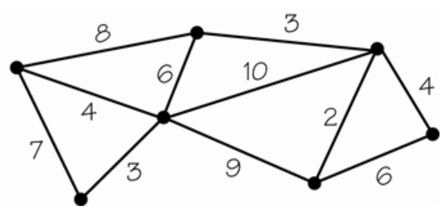
Minimum-Cost Spanning Trees

For each of the following graphs, find the minimum-cost spanning tree.

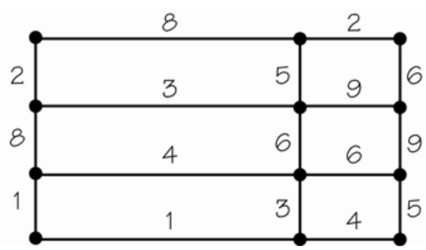
(1)



(2)



(3)



Solutions

Skills Check:

1. b, c, d
2. 90
3. a
4. 0
5. b
6. 27
7. c
8. 26
9. c
10. 32
11. c
12. *V*
13. b
14. 94
15. c
16. 72
17. b
18. 2340
19. a
20. 234
21. b
22. 14; 3
23. a
24. 7; 8; 9
25. 14
26. b
27. b
28. 18
29. c
30. 16

Collaborative Learning:

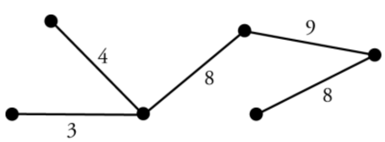
Hamiltonian Circuits:

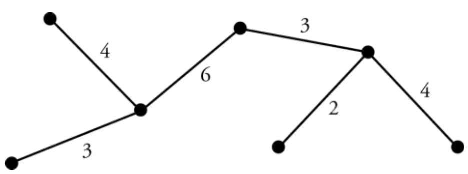
In (1), (3), and (4) there are no Hamiltonian circuits. *ABCDHGEFA* is a Hamiltonian circuit for (2).

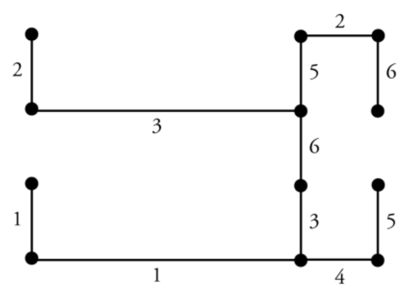
Traveling Salesman Problem:

(1) *ADBCA*—31
(2) *ABDCA*—190
(3) *ABCDEA*—30

Minimum-Cost Spanning Trees:

(1)


(2)


(3)


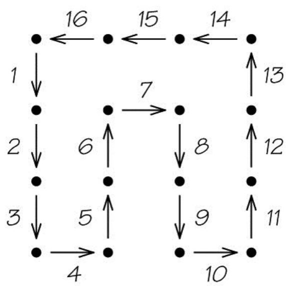
Exercises:

1. **(a)** $X_2 X_6 X_7 X_3 X_4 X_8 X_{12} X_{11} X_{10} X_9 X_5 X_1 X_2$

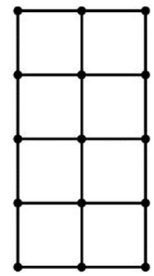
(b) $X_2 X_5 X_8 X_3 X_4 X_7 X_6 X_1 X_2$ is one possibility, as is $X_2 X_1 X_4 X_3 X_8 X_5 X_6 X_7 X_2$.

(c) $X_2 X_8 X_1 X_{10} X_7 X_6 X_9 X_5 X_4 X_3 X_2$ is one possibility, as is $X_2 X_1 X_8 X_9 X_{10} X_7 X_6 X_5 X_4 X_3 X_2$.

2. **(a)**



(b)



3. **(a)** It is not possible to make a Hamilton circuit starting at X_3.

(b) $X_3X_2X_{11}X_{12}X_6X_{13}X_{20}X_{16}X_{17}X_{14}X_{10}X_1X_4X_9X_{15}X_{18}X_{19}X_8X_7X_5X_3$

(c) $X_3X_4X_2X_5X_6X_1X_3$

4. **(a)** $X_3X_2X_5X_1X_6X_7X_8X_4X_3$

(b) 8

(c) 8

(d) 4

5. **(a)** $X_3X_6X_4X_1X_2X_5X_3$

(b) $X_3X_2X_1X_6X_7X_8X_9X_{10}X_{11}X_{12}X_5X_4X_3$

(c) $X_3X_1X_2X_7X_6X_9X_8X_5X_4X_3$

6. **(a)** $X_1X_2X_8X_{10}X_{11}X_7X_3X_4X_6X_5X_{12}X_{13}X_9X_1$

(b) Yes, $X_1X_9X_{13}X_{11}X_{10}X_8X_9X_2X_3X_4X_6X_7X_8X_2X_1X_3X_7X_{11}X_{12}X_{13}X_5X_6X_{12}X_5X_1$

(c) A Hamiltonian circuit is one where each vertex must be hit exactly one time, however not every edge must be traveled. On the other hand, an Euler circuit must have every edge traveled exactly one time, but each vertex can be crossed multiple times if needed.

7. Yes, for all three graphs

8. **(a)** A Hamiltonian circuit will remain for (a) and (b), but there will be no Hamiltonian circuit for (c).

(b) The removal of a vertex might correspond to the failure of the equipment at that site.

9. **(a)** A Hamiltonian circuit will remain for (b), but there will be no Hamiltonian circuit for (a), (c).

(b) Removing an edge between two vertices in a communications network indicates that it would no longer be possible to send messages between these two sites.

10. **(a)** **i.** Inspection of traffic lights for a small town after a storm

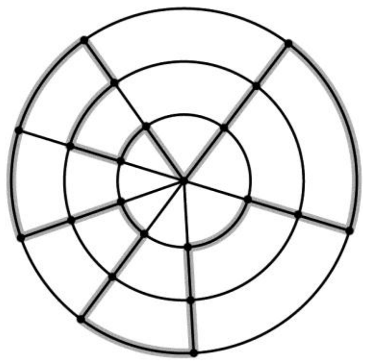
ii. Picking up medical waste at doctor offices

(b) **i**. Routing a crew to check for large tree limbs downed by the same storm

ii. Traversing the streets needed to get between the offices of the doctors in an efficient way would not seem to have any interest.

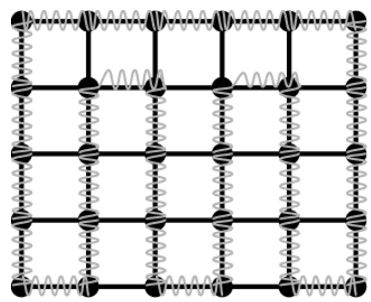
11. Other Hamiltonian circuits include *ABIGDCEFHA* and *ABDCEFGIHA.*

12. (a)



(*a*)

b.



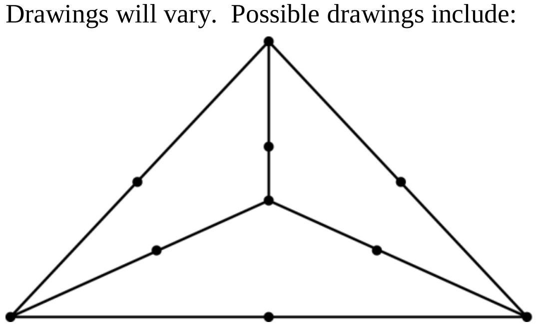
(b) A single circle that has a vertex located at the center and any number (which creates a family) of radii that have a vertex on the circle will have a Hamiltonian circuit. Additionally, any set of concentric circles (same center, but different measure of radii) will create a family of graphs if the number of radii extending from the center is even and a different family if the number of radii is odd.

For graph (b), any grid that has one side with an even number of vertices and the other side with a number of vertices (greater than 1) will form a family that has a Hamiltonian circuit.

13. (a) a. Add edge *AB*.

b. Add edge X_1X_3.

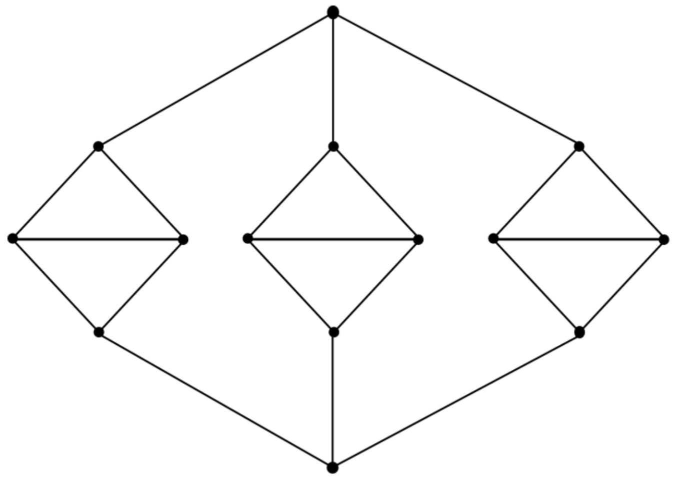
(b) Drawings will vary. Possible drawings include:



(c) Add edges X_2X_8, X_8X_6, X_6X_4, and X_4X_2.

14. A Hamiltonian circuit must use edges *AD*, *DE*, *BE*, and *AB*. These edges already form a circuit making it impossible to visit *C* as part of the circuit.

15. The graph below has no Hamiltonian circuit, and every vertex of the graph has valence 3.



16. The "circuit" given repeats vertices *B* and *C*. There is no Hamiltonian circuit in this graph.

17. (a) Any Hamiltonian circuit would have to use both edges at the vertices X_5, X_4, and X_2. This would cause a problem in the way a Hamiltonian circuit could visit vertex X_1. Thus, no Hamiltonian circuit exists.

(b) If there were a Hamiltonian circuit, it would have to use the edges X_4 and X_5 and X_6 and X_7. This would make it impossible for the Hamiltonian circuit to visit X_8 and X_9. Thus, no Hamiltonian circuit exists.

18. (a) $X_6X_3X_7X_5X_2X_1X_4X_6$ is a Hamiltonian circuit.

(b) There still is no Hamiltonian circuit.

19. (a) Yes, there is a Hamilton circuit.

(b) No Hamiltonian circuit.

(c) No Hamiltonian circuit.

20. (a) For any $m = 2$ and $n \geq 1$, the graph has a Hamiltonian circuit.

(b) If either *m* or *n* is odd, the graph has a Hamiltonian circuit. If both *m* and *n* are even, the graph has no Hamiltonian circuit. A real-world application would be to design an efficient route to check that the traffic control equipment at each vertex was in proper working order.

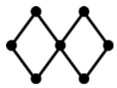
For grid graphs where there are an even number of blocks on each side, there is no Hamiltonian circuit. Yet, by repeating only one edge and vertex one can find a tour that visits all the other vertices in such a grid graph exactly once.

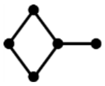
21. (a) There is a Hamiltonian path from X_3 to X_4.

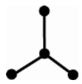
(b) No. There is a Hamiltonian path from X_1 to X_8 in graph (b).

(c) Here are two examples: A worker who inspects sewers may start at one garage at the start of the work day but may have to report to a different garage for the afternoon shift. A school bus may start at a bus garage and then pick up students to take them to school, where the bus sits until the end of the school day.

22. (a)

(b) 

(c) 

(d) 

23. (a) Hamiltonian circuit, yes. One example is $X_1X_3X_7X_5X_6X_8X_2X_4X_1$. Euler circuit, yes.

(b) Hamiltonian circuit, yes. One example is $Y_1Y_3Y_2Y_5Y_6Y_4Y_1$. Euler circuit, yes.

(c) Hamiltonian circuit, yes; Euler circuit, no. One example is:

$Z_2Z_{11}Z_5Z_4Z_6Z_{10}Z_{16}Z_{12}Z_{13}Z_{15}Z_9Z_7Z_3Z_8Z_{14}Z_1Z_2$.

(d) Hamiltonian circuit, no; Euler circuit, yes. One example is:

$$U_1U_2U_5U_6U_{16}U_{15}U_{11}U_4U_5U_{12}U_{11}U_{10}U_{14}U_{13}U_7U_3U_8U_{10}U_9U_3U_1.$$

24. (a) You can get an *n*-cube by taking two copies of the $(n-1)$-cube. Given that there is a Hamilton circuit in each $(n-1)$-cube, then you may remove any single corresponding edge of each of these $(n-1)$-cubes. Connect the corresponding vertices by adding edges between the two $(n-1)$-cubes. You have an *n*-cube, and this *n*-cube will have a Hamilton circuit.

(b) The *n*-cube has 2^n vertices, and the number of edges of the *n*-cube is equal to twice the number of edges of an $(n-1)$-cube plus 2^{n-1}. A formula for this number is $n2^{n-1}$.

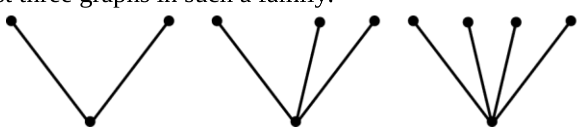
25. (a) Hamiltonian circuit, yes; Euler circuit, no.

(b) Hamiltonian circuit, yes; Euler circuit, no.

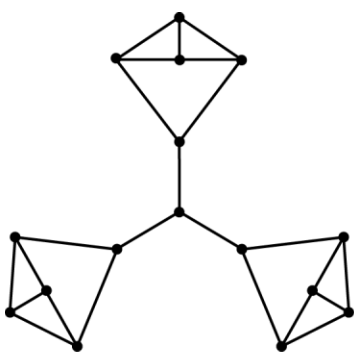
(c) Hamiltonian circuit, yes; Euler circuit, no.

(d) Hamiltonian circuit, no; Euler circuit, no.

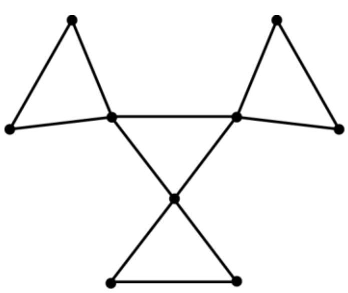
26. The diagram shows the first three graphs in such a family.



27. (a)



(b)



(c) A graph has an Euler path if for two different vertices *u* and *v* of the graph there is a path from *u* to *v* that uses each edge of the graph once and only once.

28. **(a)** $4 \times 5 \times 4 \times 3 = 240$
(b) $3 \times 1 \times 4 \times 3 = 36$

29. **(a)** $7 \times 6 \times 5 \times 4 \times 3 = 2520$
(b) $7 \times 7 \times 7 \times 7 \times 7 = 16{,}807$
(c) $7^5 - 7 = 16{,}800$

30. For this exercise, letters are considered case sensitive.
$52 \times 5 \times 4 = 1040$

31. **(a)** $9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 = 15{,}120$
(b) $(26)(26)(26) = 17{,}576$

32. $(12-2)\times(6-3) = 10 \times 3$ = 30 weeks.

33. The new system is an improvement since it codes 676 locations compared with 504 for the old system. This is 172 more locations.

34. There are five ways to choose the position into which to put the note that is to be sharped, and seven ways to fill this position with a note. Since there is no repetition, the remaining four positions can be filled in $6 \times 5 \times 4 \times 3$ ways, for a total of $5 \times 7 \times 6 \times 5 \times 4 \times 3 = 12{,}600$ ways to create the logo.

35. **(a)** $(26)(26)(26)(10)(10)(10) - (26)(26)(26) = 26^3\left(10^3 - 1\right) = 17{,}558{,}424$
(b) Answers will vary.

36. The number of different meal choices is $6(10)(8) = 480$. The number of choices avoiding pie as a dessert is $6(10)(5)$ = 300.

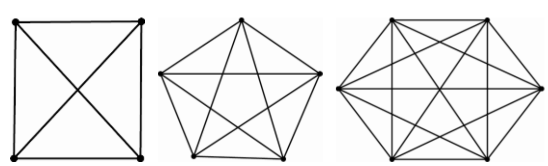
37. With no other restrictions, $10^7 = 10{,}000{,}000$. With no other restrictions, $9 \times 10^2 = 900$.

38. **(a)** We can apply the fundamental principle of counting in the following way:
Step 1: Pick one of the four positions to put the 0. This can be done in four ways.
Step 2: Pick one of the remaining 9 digits to repeat. This can be done in 9 ways.
Step 3: Pick two positions out of the remaining three to put the repeated digit into. This can be done in three ways.
Step 4. Fill the fourth position with a digit different from 0 and the one used in Step 2. This can be done in eight ways.

Hence, there are $4(9)(3)(8) = 864$.

(b) $10(10)(10)(10) = 10{,}000$.

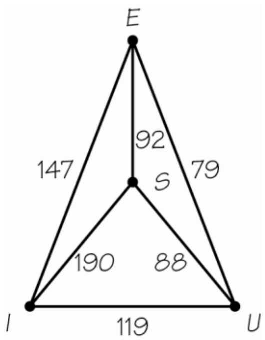
39. The graphs below (left to right) have 6, 10, and 15 edges.



The n-vertex complete graph has $\frac{n(n-1)}{2}$ edges. The number of TSP tours is 3, 12, and 60, respectively.

40. 5! = 120; 6! = 720; 7! = 5040; 8! = 40,320; 9! = 362,880; 10! = 3,628,800. The number of TSP tours in a complete graph on n vertices is $\frac{(n-1)!}{2}$. Thus for eight vertices we have $\frac{(8-1)!}{2}=\frac{7!}{2}=\frac{5,040}{2}=2520.$

41. (a)



(b) (1) *UISEU;* mileage $=119+190+92+79=480$
(2) *USIEU;* mileage $=88+190+147+79=504$
(3) *UIESU;* mileage $=119+147+92+88=446$

(c) *UIESU* (Tour 3)

(d) No

(e) Starting from *U*, one gets *UESIU* Tour 1. From *S* one gets *SUEIS* Tour 2, from *E* one gets *EUSIE* Tour 2, and from *I* one gets *IUESI* Tour 1.

(f) *EUSIE* Tour 2; No

42. *FRCMF* is quickest and takes 40 minutes.

43. *FCMRF* gets her home in 42 minutes.

44. *FMCR* is quickest and takes 33 minutes.

45. (a) Using the nearest-neighbor algorithm, the route would be *HBCAH*.

(b) Using the sorted-edges algorithm, the route would be *HBCAH*.

(c) There are $\frac{(4-1)!}{2}=\frac{3!}{2}=\frac{3\times2\times1}{2}=\frac{6}{2}=3$ Hamilton circuits to examine.

The circuits and their weights are as follows:

HABCH weight $=35+33+28+37=133$

HCABH weight $=37+39+33+30=139$

HBCAH weight $=30+28+39+35=132$

The best route in terms of time is 132. This coincides with both the nearest-neighbor and sorted-edges solutions from *H*.

46. (a) Using the nearest-neighbor algorithm, the route would be *HBCAH*.

(b) Using the sorted-edges algorithm, the route would be *HBCAH*.

(c) There are $\frac{(4-1)!}{2}=\frac{3!}{2}=\frac{3\times2\times1}{2}=\frac{6}{2}=3$ Hamilton circuits to examine.

The circuits and their weights are as follows:

HABCH weight $=39+37+32+41=149$

HCABH weight $= 41 + 43 + 37 + 34 = 155$

HBCAH weight $= 34 + 32 + 43 + 39 = 148$

The best route in terms of time is 148. This coincides with both the nearest-neighbor and sorted-edges solutions from *H*.

(d) Each of the times in the diagram for this exercise are exactly 4 units longer than the ones in Exercise 45. Additionally, the circuits in this exercise are 16 units longer than those in Exercise 45, which, of course, makes perfect sense since there are four edges to traverse. Hence, if one changes the length of the edges by a factor of *n*, then the weight of each circuit will change by the number of edges traversed times *n*.

47. *MACBM* takes 344 minutes to traverse.

48. **(a)** The two methods give identical answers for the first graph. It is *BAEDCB*. In the second graph nearest-neighbor yields *BDAECFB*, while sorted-edges yields *BDFCEAB*.

(b) If a complete graph has *n* vertices, then there are $\frac{(n-1)!}{2}$ Hamiltonian circuits. The first graph would have $\frac{(5-1)!}{2} = \frac{4!}{2} = \frac{4\times3\times2\times1}{2} = \frac{24}{2} = 12$ Hamilton circuits to examine. The second graph would have $\frac{(6-1)!}{2} = \frac{5!}{2} = \frac{5\times4\times3\times2\times1}{2} = \frac{120}{2} = 60$ Hamilton circuits to examine.

(c) Answers will vary.

49. This is an example of a traveling salesman problem.

50. **(a)** *ACBDA* is both the nearest-neighbor and sorted-edges tour.

(b) Nearest-neighbor: *ABCDA* cost 1170; sorted-edge: *ABDCA* cost 1020

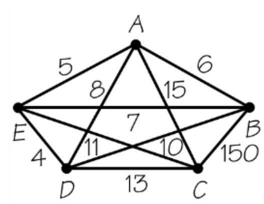
(c) *ADBCEA* is the nearest-neighbor tour; *ADEBCA* is the sorted-edges tour.

51. A sewer drain inspection route at corners involves finding a Hamiltonian circuit, and there is such a circuit. If the drains are along the blocks, a route in this case involves solving a Chinese postman problem. Since there are 18 odd-valent vertices, an optimal route would require at least 9 reuses of edges. There are many such routes that achieve 9 reuses.

52. **(a)** *AFEDCBA* (from *A*); *BFACDEB* (from *B*)

(b) Sorted-edges: *AFEDBCA*

53. The complete graph shown has a different nearest-neighbor tour that starts at *A* (*AEDBCA*), a sorted-edges tour (*AEDCBA*), and a cheaper tour (*ADBECA*).



54. There would be $\frac{(20-1)!}{2} = 6.1\times10^{16}$ Hamiltonian circuits whose cost would have to be computed. This would take 1.9 years at a billion tours per second.

55. The optimal tour is the same, but its cost is now $4200 + 10(50) = 4700$.

56. **(a)** The graph shown is not a tree because it contains a circuit.

(b) The graph shown is not a tree because it contains a circuit.

(c) The graph shown is a tree.

(d) The graph shown is a tree.

(e) The graph shown is not a tree because it is not connected.

(f) The graph shown is not a tree because it is not connected.

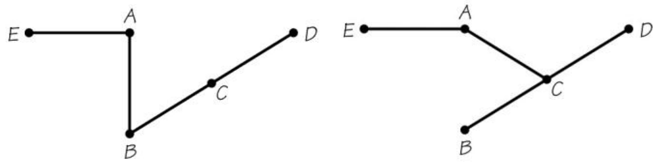
57. (a) a. Not a tree because there is a circuit. Also, the wiggled edges do not include all vertices of the graph.
b. The circuit does not include all the vertices of the graph.

(b) a. The tree does not include all vertices of the graph.
b. Not a circuit

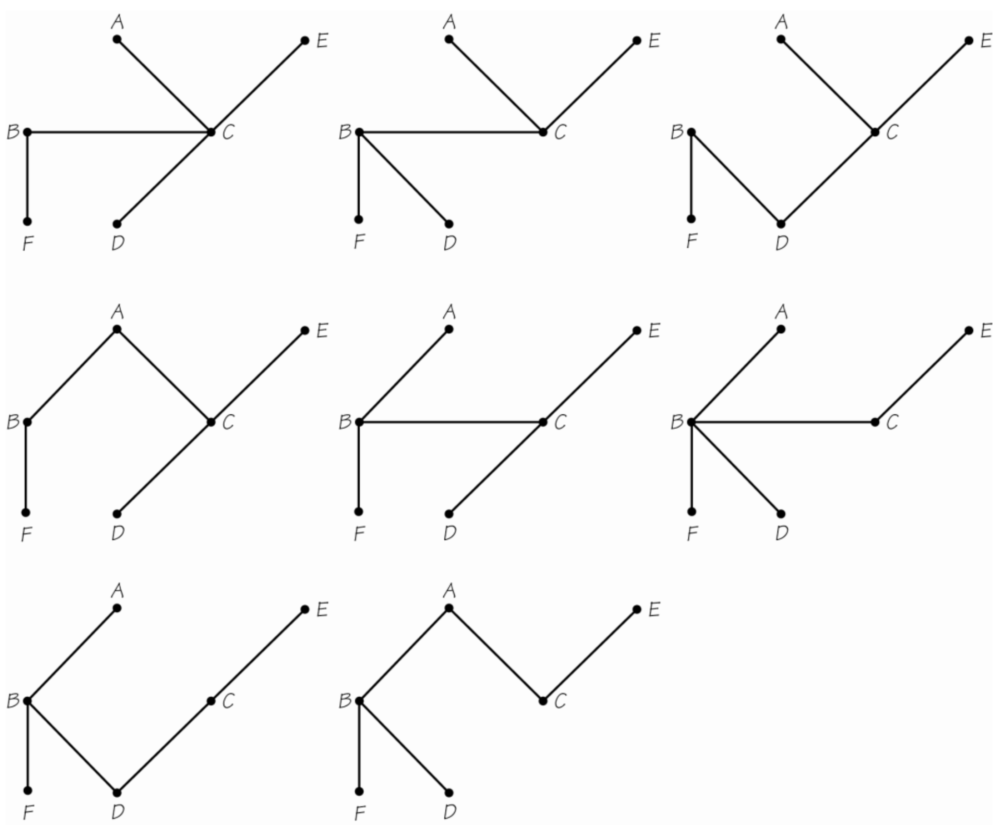
(c) a. Not a tree
b. Not a circuit

(d) a. Not a tree
b. Not a circuit

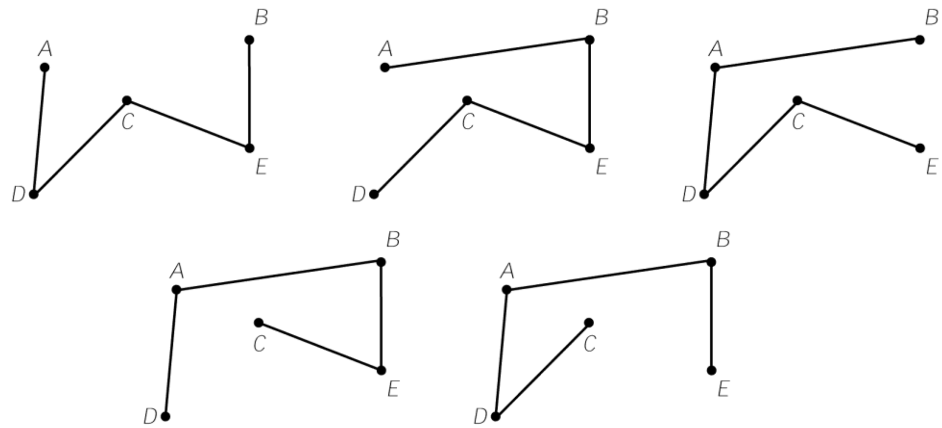
58. (a)



(b)



(c)



59. **(a)** 1, 2, 3, 4, 5, 8; cost is 23

(b) 1, 1, 1, 2, 2, 3, 3, 4, 5, 6, 6; cost is 34

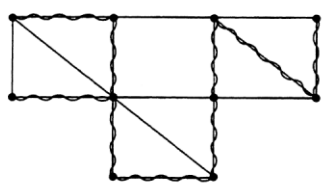
(c) 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 5, 5, 6, 7; cost is 60

(d) 1, 2, 2, 3, 3, 3, 4, 5, 5, 5, 6, 6; cost is 45

60. If *G* is connected and has 14 vertices, a spanning tree of *G* will have 13 edges. Any spanning tree of *G* will have 14 vertices. One can conclude that *G* must have at least 13 edges (when *G* itself is a tree), but nothing more.

61. The spanning tree will have 27 vertices. *H* also will have 27 vertices. The exact number of edges in *H* cannot be determined, but *H* has at least 26 edges.

62. The wiggled edges in the figure constitute a spanning tree whose cost is minimal.



63. Yes

64. Examples include the synthesis of the links to create a wireless communications network or a homeowner putting underground sprinkler pipes into a large garden.

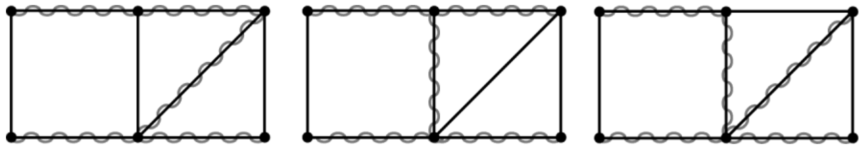
65. Yes. Change all the weights to negative numbers and apply Kruskal's algorithm. The resulting tree works, and the maximum cost is the negative of the answer you get. If the numbers on the edges represent subsidies for using the edges, one might be interested in finding a maximum-cost spanning tree.

66. Air distances may yield a different solution.

67. A negative weight on an edge is conceivable, perhaps a subsidization payment. Kruskal's algorithm would still apply.

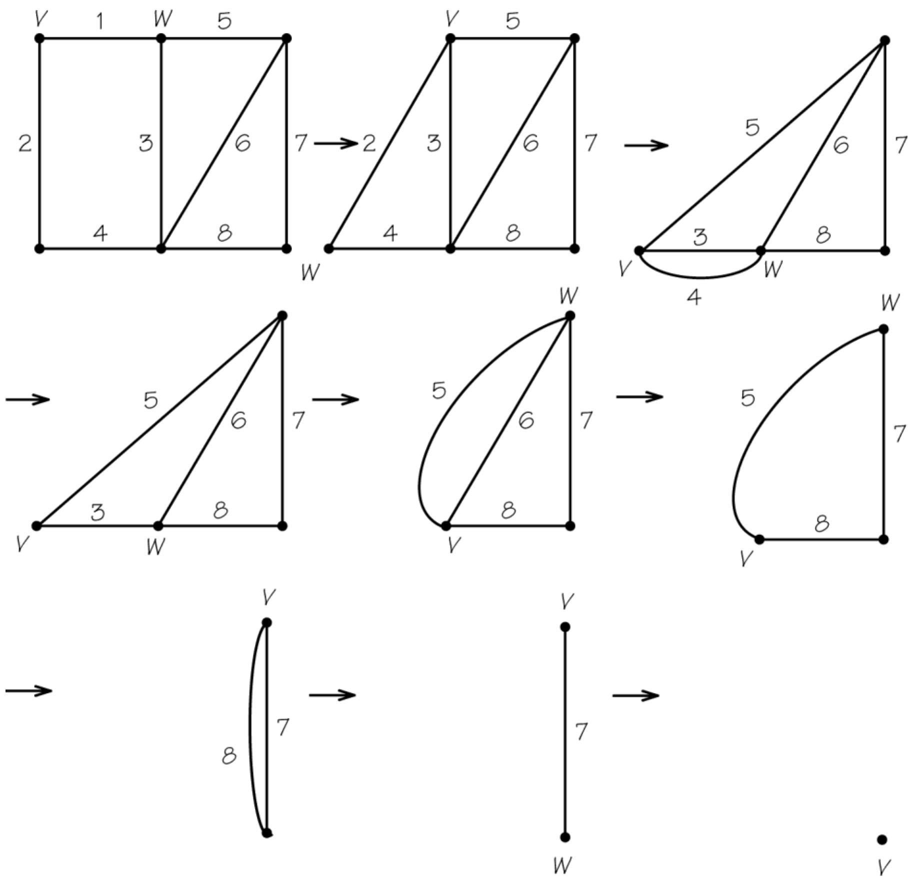
68. Kruskal's algorithm works by at each stage selecting the cheapest edge not already selected which does not form a circuit with the edges already chosen. If all the weights on the edges are different from each other, then when a decision is made as to which edge to add next to those already selected, there will never be a tie. Thus, the choice at each stage is always unique, and, hence, there can only be one minimum-cost spanning tree. If at some stage one cannot add an edge to those already chosen because a circuit would form, this makes no difference, because the next edge that can be added will never involve a choice among edges with the same weight.

69. Three different trees with the same cost are shown:

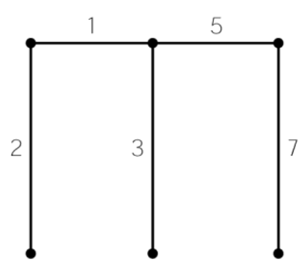


Weight = 2+2+2+4+5=15 Weight = 2+2+2+4+5=15 Weight = 2+2+2+4+5=15

70. (a)–(c)

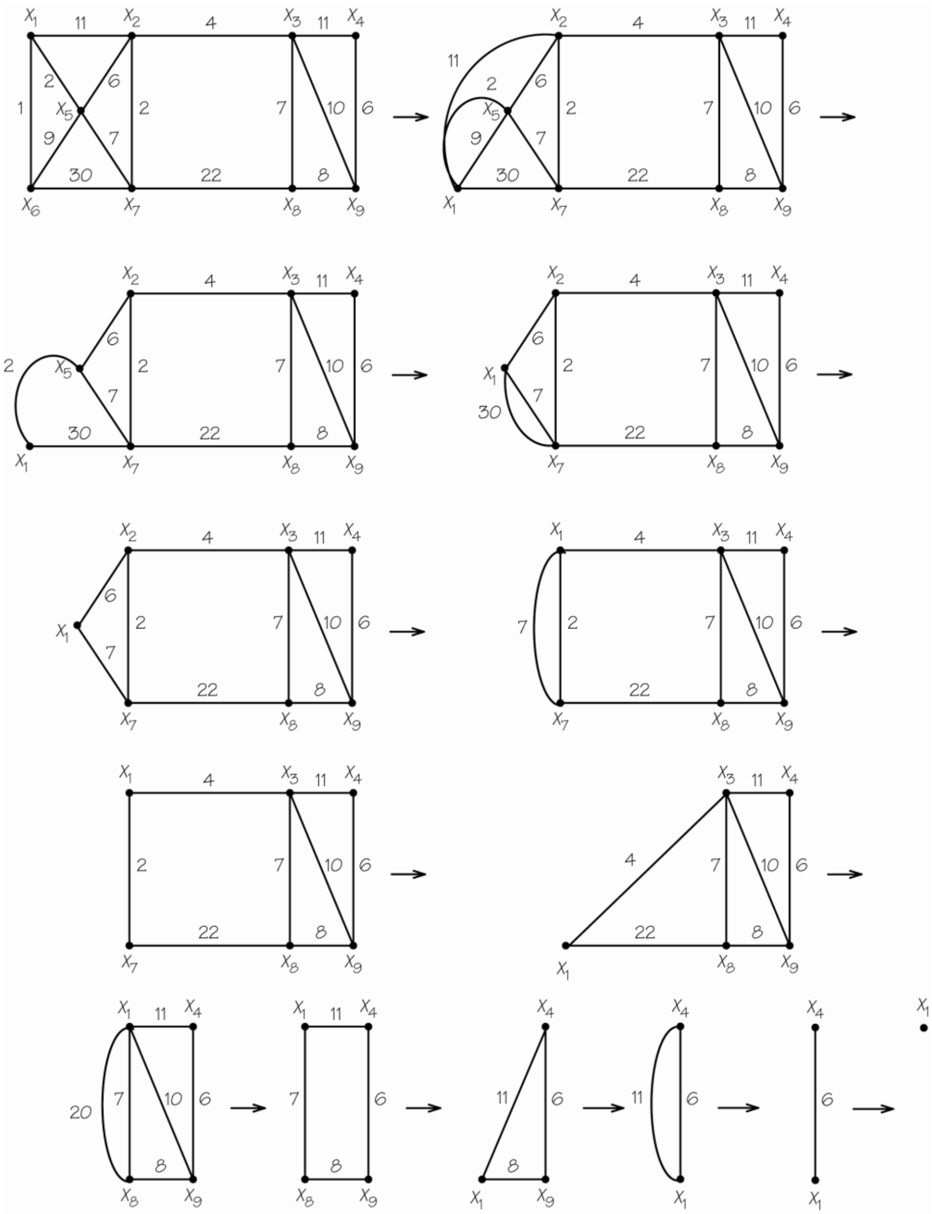


The minimum-cost spanning tree is as follows:

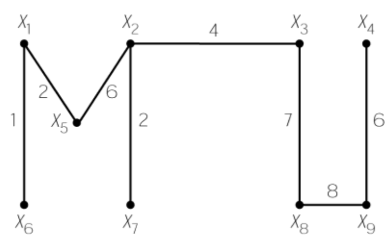


(d) a.

If one starts at vertex X_1, the sequence is as follows:

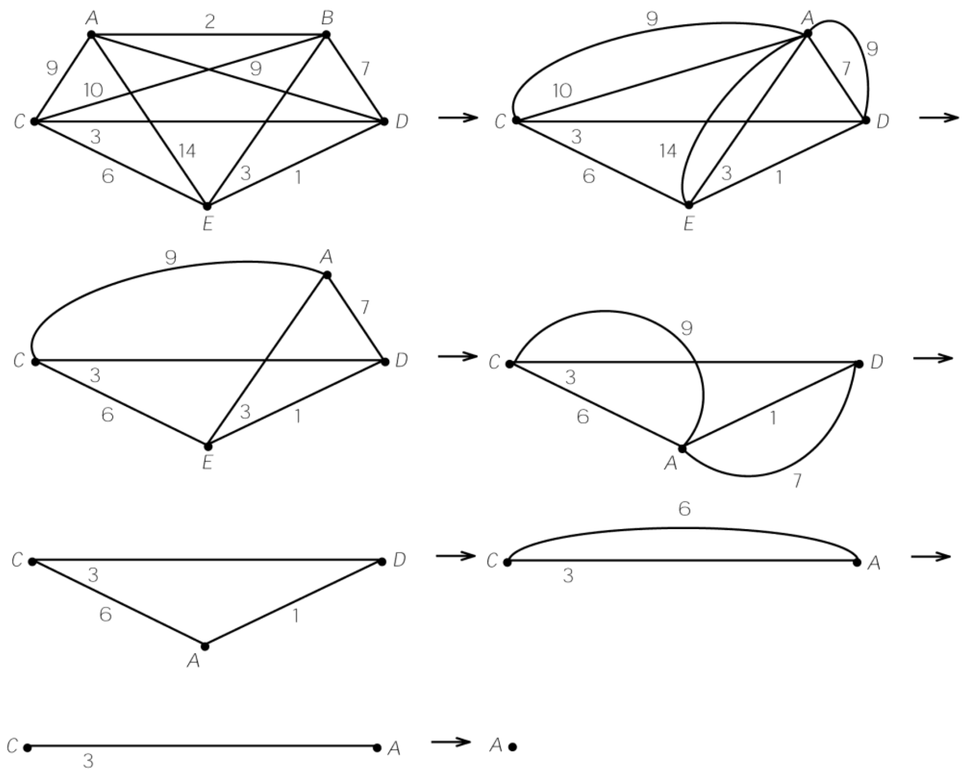


The minimum-cost spanning tree is as follows:

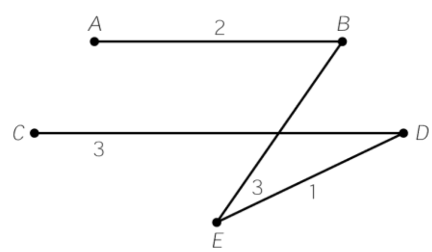


(*continued on next page*)

(d) (*continued*)
b.



The minimum-cost spanning tree is as follows:



71. **(a)** True
(b) False (unless all the edges of the graph have the same weight)
(c) True
(d) False
(e) False

72. **(a)** Starting at the weight-5 vertex, the cost is $5+(4+8+7)=24$.
Starting at the weight-10 vertex, the cost is $10+(13+4+2)=29$.
Starting at the weight-9 vertex, the cost is $9+(2+7+11)=29$.
Starting at the weight-6 vertex, the cost is $6+(13+8+11)=38$.
The minimum cost would be 24.
(b) Starting at the weight-12 vertex, the cost is $12+(15+2+7)=36$.

Starting at the weight-3 vertex, the cost is $3+(2+14+9)=28.$

Starting at the weight-28 vertex, the cost is $28+(7+9+4)=48.$

Starting at the weight-9 vertex, the cost is $9+(15+14+4)=42.$

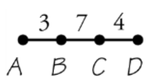
The minimum cost would be 28.

If the vertex costs are neglected, the minimum-cost solution in graph (a) is $2+4+8=14.$ Similarly, the minimum-cost solution in graph (b) is $2+4+7=13.$

73. (a) Answers will vary for each edge, but the reason it is possible to find such trees is that each edge is an edge of some circuit.

(b) The number of edges in every spanning tree is five, one less than the number of vertices in the graph.

(c) Every spanning tree must include the edge joining vertices *C* and *D*, since this edge does not belong to any (simple) circuit in the graph. Additionally, every spanning tree must also include the edge joining A and G in order to stay connected.

74. (a) 

(b) The vertices in the graph might represent locations along a road, and the distances between the locations are given by the table shown. Distances along a road would naturally be represented by a graph, which is a path. Alternatively, the vertices in the graph might represent manuscripts that were copied by hand from other manuscripts. The weights in the table in this case might represent numbers of key sentences where the manuscripts differ. The graph representing the table, in this case being a path, suggests that each manuscript was copied from a "prior" manuscript, rather than two manuscripts being copied from one common ancestor, say.

75.

	A	B	C	D
A	0	16	13	5
B	16	0	19	11
C	13	19	0	8
D	5	11	8	0

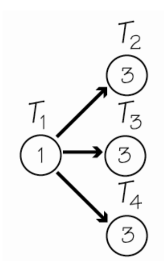
76. (a) The earliest completion time is 38 because the longest path, the unique critical path $T_1 T_4 T_7$, has length 38.

(b) The earliest completion time is 38 because the longest path, the unique critical path $T_1 T_3 T_5 T_8$, has length 38.

77. (a) The earliest completion time is 22 because the longest path, the unique critical path $T_3 T_2 T_5$, has length 22.

(b) The earliest completion time is 30 because the longest path, the unique critical path $T_3 T_5 T_7$, has length 30.

78.

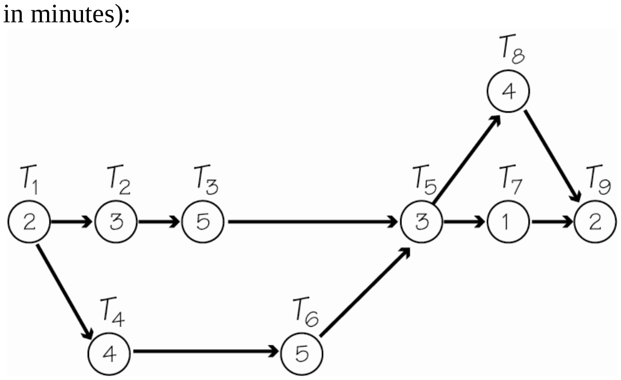


79. The only tasks which if shortened will reduce the earliest completion time are those on the critical path, so in this case, these are the tasks T_1, T_5, and T_7. If T_5 is shortened to 7, then the longest path will have length 29, and this becomes the earliest completion time. The tasks on this critical path are T_1, T_4, and T_7.

80. The critical path is T_1, T_5, T_7 with length 31. If T_5's time is reduced by 2, T_1, T_4, and T_7 will now also be a critical path, and the new lengths of both of these paths will be 29.

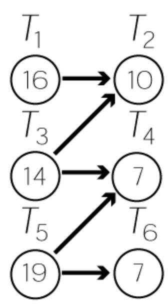
81. Different contractors will have different times and order-requirement digraphs. However, in any sensible order-requirement digraph, the laying of the foundation will come before the erection of the side walls and the roof. The fastest time for completing all the tasks will be the length of the longest path in the order-requirement digraph.

82. One possibility is (times in minutes):



The earliest completion time is 19 minutes.

83. One example is given below.



84. Using the nearest-neighbor and the sorted-edges method, the shortest way to deliver all packages from D would be DEABC with a length of 31.

85. Mary has 26 × 26 × 25 × 3 × 9 × 8 × 7 × 26 = 664,372,800 different passwords!

86. Yes. One such circuit is: *HICFDJABEGH*.

87. Using Kruskal's algorithm, the edges are added to the minimum-cost spanning tree in this order: *GF*, *JD*, *IH*, *KL*, *HK*, *IC*, *JL*, *FL*, *LE*, *AI*, *AB*.

Word Search Solution