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 Fig. 2.2 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 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 *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 Hamiltonian circuits in a graph with a given number of vertices.

6. Define 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 not 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 the text example on page 41 and 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 students have probably 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 they are named after (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.

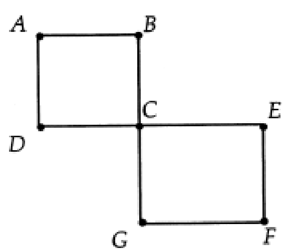
Minimum-Cost Spanning Trees

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.

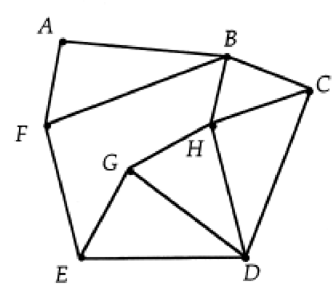
Hamiltonian Circuits

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

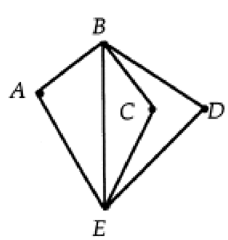
(1)



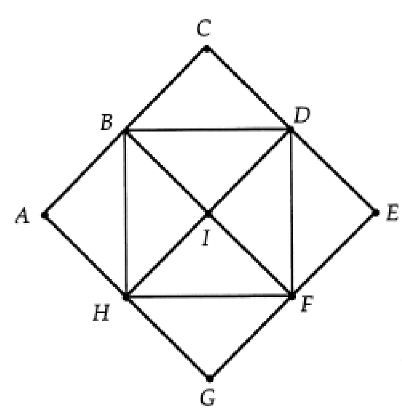
(2)



(3)



(4)

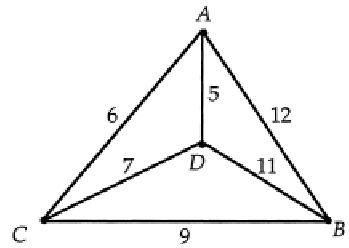


2. Use the campus map as in the exercise from Chapter 1, but now in the context of Hamiltonian circuits. If you wish, you can include distances and turn this into a Traveling Salesman's Problem.

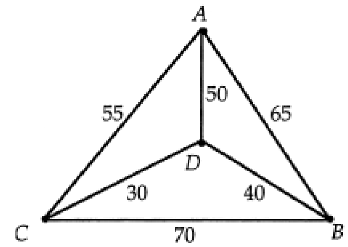
Traveling Salesman Problem

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

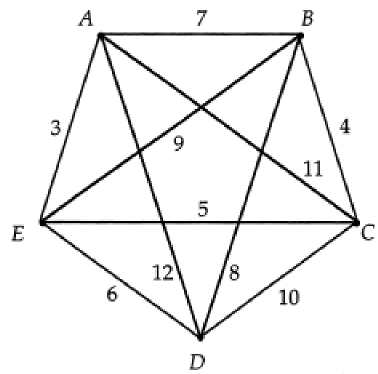
(1)



(2)



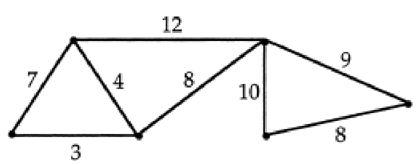
(3)



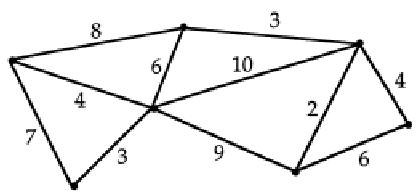
Minimum-Cost Spanning Trees

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

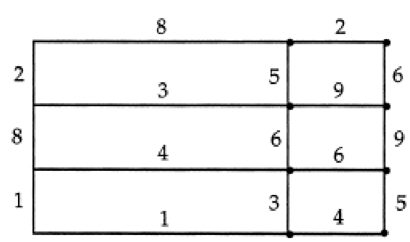
(1)



(2)



(3)



Solutions

Skills Check:

1. c	2. b	3. c	4. b	5. b	6. b	7. c	8. c	9. a	10. c
11. b	12. a	13. b	14. b	15. c	16. a	17. b	18. c	19. a	20. c

Cooperative Learning:

Hamiltonian Circuits:

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

Traveling Salesman Problem:

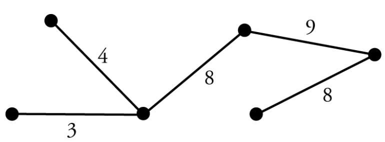
(1) *ADCBA*—33

(2) *ABDCA*—190

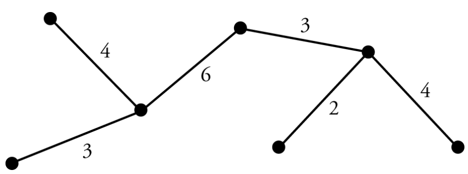
(3) *ABCDEA*—30

Minimum-Cost Spanning Trees:

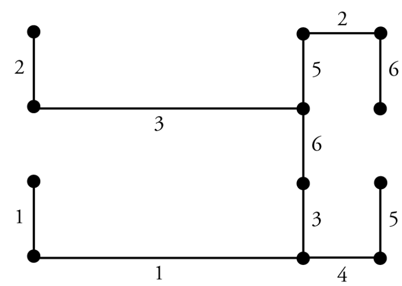
(1)



(2)



(3)



Exercises:

1. (a) $X_5 X_6 X_1 X_3 X_4 X_2 X_5$

(b) $X_5 X_4 X_3 X_2 X_1 X_6 X_7 X_8 X_9 X_{10} X_{11} X_{12} X_5$

(c) $X_5 X_4 X_3 X_1 X_2 X_7 X_6 X_9 X_8 X_5$

(d) $X_5 X_8 X_3 X_4 X_7 X_6 X_1 X_2 X_5$

(e) $X_5 X_4 X_3 X_2 X_8 X_1 X_{10} X_7 X_6 X_9 X_5$

2. (a) Yes.

(b) Yes.

(c) Yes.

(d) Yes

(e) Yes.

(f) Yes

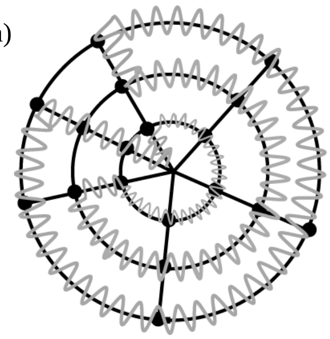
3. (a) A Hamiltonian circuit will remain for (a) and (b), but there will be no Hamiltonian circuit for (c), (d), and (e),

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

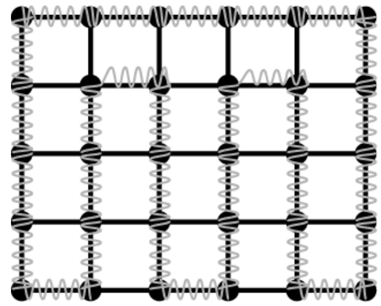
4. Finding a Hamiltonian circuit in a graph would model running a collection of errands starting and ending at a dorm room, inspecting storm sewers after a storm, and a sequence of operations on individual cars that a robot on an automobile assembly line would need to carry out.

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

6. (a)



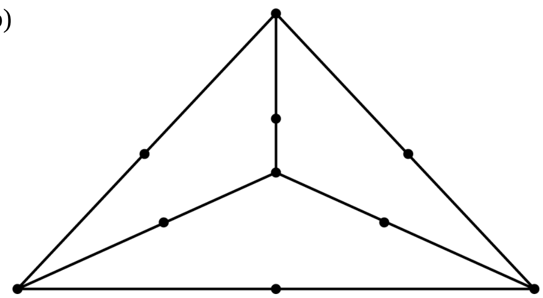
(b)



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

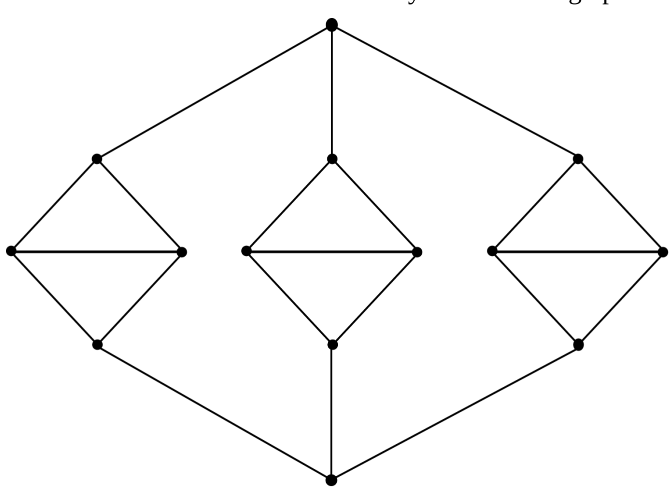
b. Add edge X_1X_3.

(b)



8. 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.

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



10. Any Hamiltonian circuit must use edge *BC*, but removal of this edge disconnects the graph into two components consisting of circuits. This precludes there being an edge in addition to *BC* to be part of a Hamiltonian circuit.

11. (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.

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

(b)There still is no Hamiltonian circuit.

13. (a) No Hamilton circuit.

(b) No Hamilton circuit.

(c) No Hamilton circuit.

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

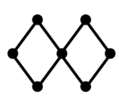
(b) If either *m* or *n* is odd, the graph has a Hamiltonian circuit. If both *m* and *n* are even, the graph has no Hamiltonian circuit. A real-world application would be to design an efficient route to check that the traffic control equipment at each vertex was in proper working order. For grid graphs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

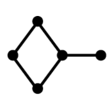
15. (a) There is a Hamiltonian path from X_3 to X_4.

(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.

16. (a)

(b) 

(c) 

17. (a) Hamiltonian circuit, yes. One example is: $X_1X_3X_7X_5X_6X_8X_2X_4X_1$; Euler circuit, no.

(b) Hamiltonian circuit, yes. Euler circuit, yes.

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

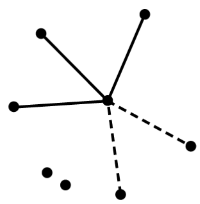
(d) Hamiltonian circuit, no. Euler circuit, yes. One example is as follows.

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

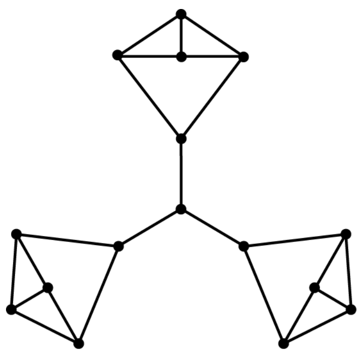
18. The *n*-cube has 2^n vertices, and the number of edges of the *n*-cube is equal to twice the number of edges of an $(n-1)$-cube plus 2^{n-1}. A formula for this number is $n2^{n-1}$.

19. (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.

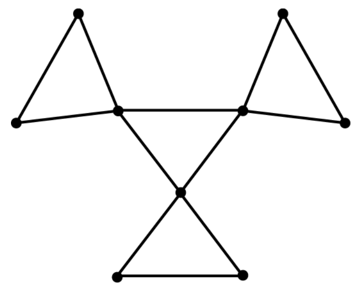
20.



21. (a)



(b)



(c) A graph has an Eulerian 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.

22. $(12-2)\times(6-3)=10\times3$ = 30 weeks.

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

24. $52\times99\times98\times97\times96\times95$

25. (a) $9\cdot8\cdot7\cdot6\cdot5=15{,}120$
(b) $(26)(26)(26)=17{,}576$

26. (a) $3 \times 4 \times 4 \times 3 = 144$
(b) $3 \times 1 \times 4 \times 3 = 36$

27. (a) $7\times6\times5\times4\times3=2520$
(b) $7\times7\times7\times7\times7=16{,}807$
(c) $7^5-7=16{,}800$

28. 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 $6\times5\times4\times3$ ways for a total of $5\times7\times6\times5\times4\times3=12{,}600$ ways to create the logo.

29. (a) $(26)(26)(26)(10)(10)(10)-(26)(26)(26)=26^3(10^3-1)=17{,}558{,}424$
(b) Answers will vary.

30. The number of different meal choices is $4(10)(8)=320$. The number of choices avoiding pie as a dessert is $4(10)(5)$ = 200.

31. With no other restrictions, $10^7=10{,}000{,}000$. With no other restrictions, $9\times10^2=900$.

32. (a) We can apply the fundamental principle of counting in the following way.
Step 1: Pick one of the 4 positions to put the 0. This can be done in 4 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 8 ways.

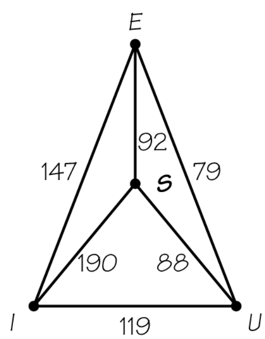
Hence, there are $4(9)(3)(8)=864$.

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

33. These graphs have 6, 10, and 15 edges, respectively. The *n*-vertex complete graph has $\frac{n(n-1)}{2}$ edges. The number of TSP tours is 3, 12, and 60, respectively.

34. 5! = 120; 6! = 720; 7! = 5,040; 8! = 40,320; 9! = 362,880; 10! = 3,628,800. The number of TSP tours in a complete graph on 9 vertices is 20,160.

35. (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.

36. *FMCRF* is quickest and takes 36 minutes.

37. *FMCRF* gets her home in 36 minutes.

38. *FMCR* is quickest and takes 30 minutes.

39. *MACBM* takes 344 minutes to traverse.

40. (a) The two methods give identical answers in this case: *BAEDCB*.

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

(c) Answers will vary.

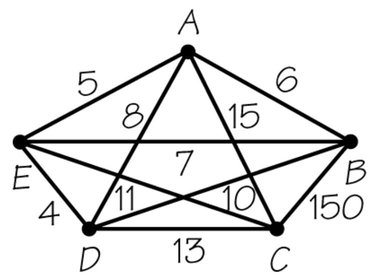
41. A traveling salesman problem.

42. (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.

43. 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.

44. (a) *AFEDCBA* (from *A*); *BFACDEB* (from *B*)
(b) Sorted edges: *AFEDBCA*

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



46. 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.

47. The optimal tour is the same but its cost is now $4200+10(30)=4500$.

48. (a) The graph shown is not a tree because it contains a circuit.
(b) The graph shown is a tree.
(c) The graph shown is not a tree because it contains a circuit.
(d) The graph shown is not a tree because it is not connected.
(e) The graph shown is a tree.
(f) The graph shown is not a tree because it is not connected.
(g) The graph shown is a tree.

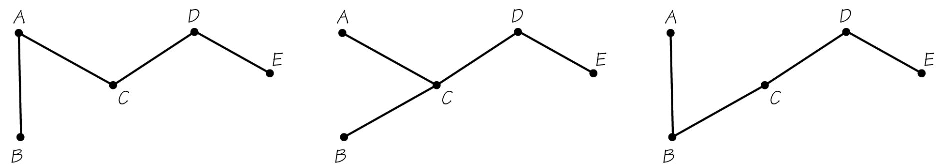
49. (a) a. Not a tree because there is a circuit. Also, the wiggled edges do not include all vertices of the graph.
b. The circuit does not include all the vertices of the graph.

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

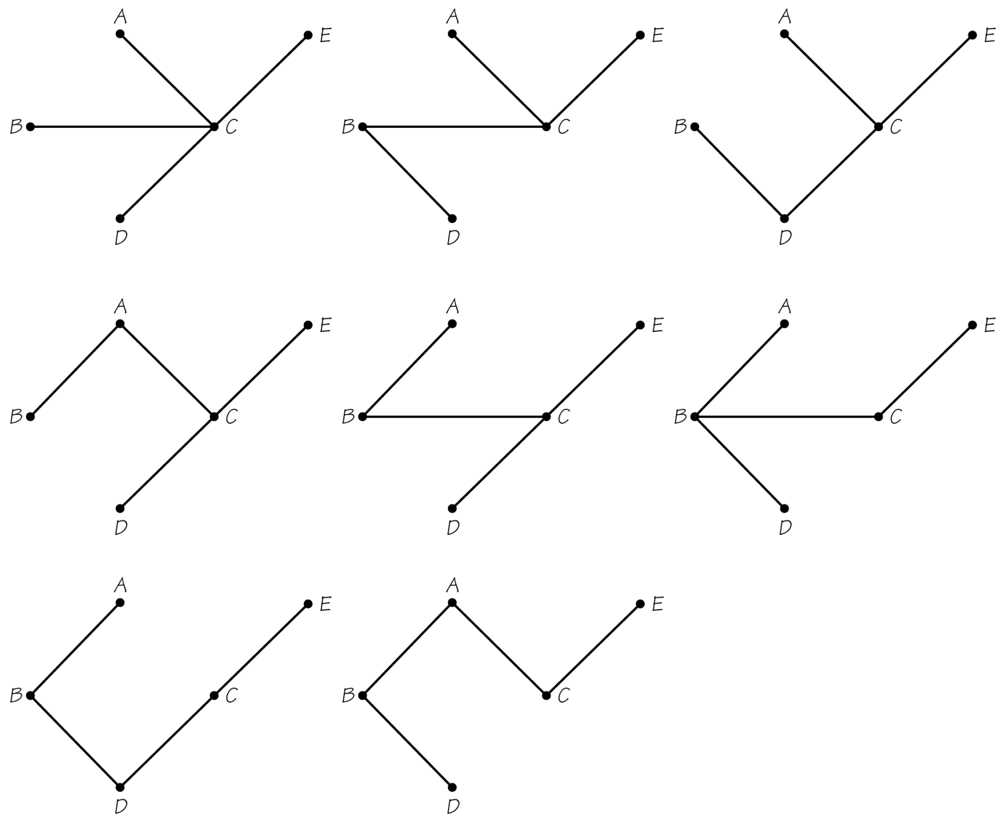
(c) a. Not a tree.
b. Not a circuit.

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

50. (a)



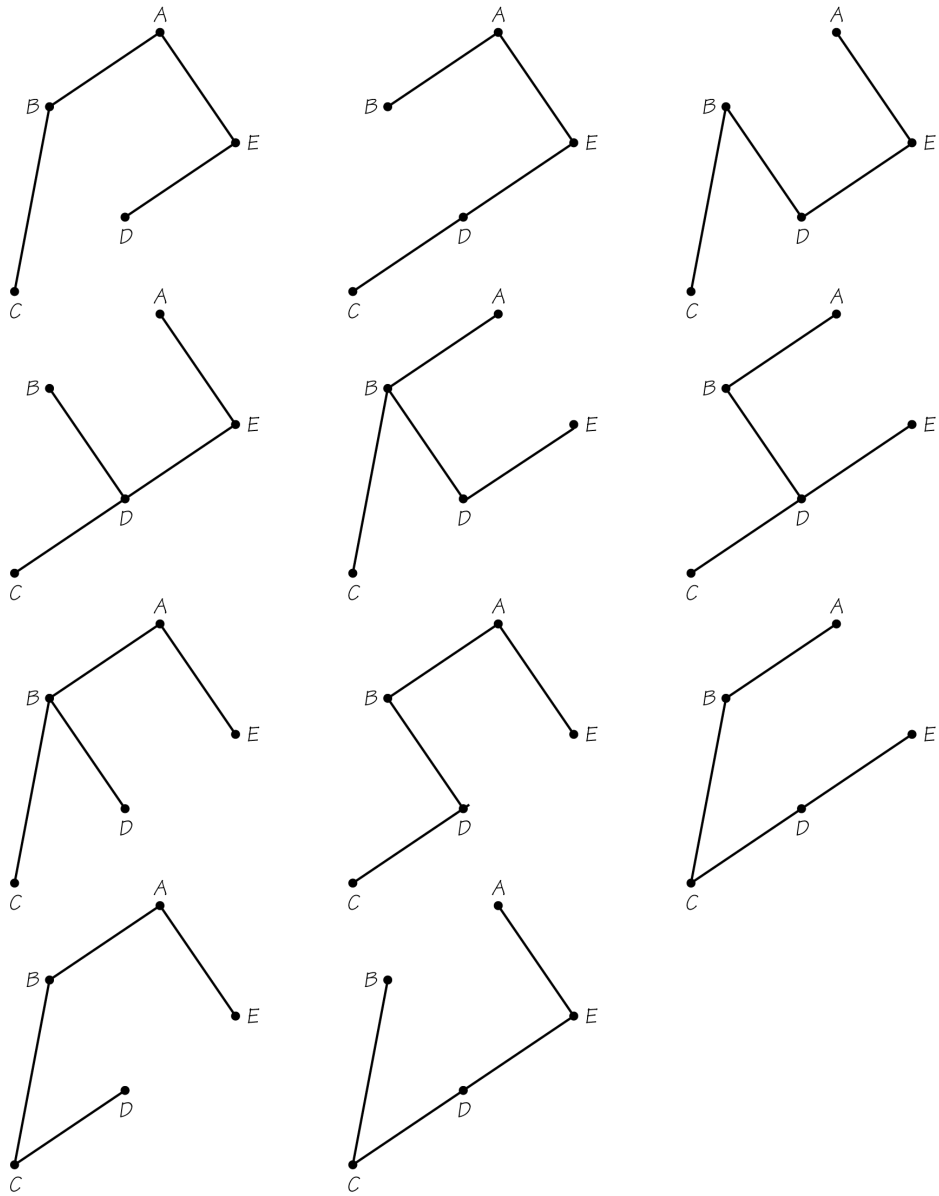
(b)



Continued on next page

50. continued

(c)



51. (a) 1, 2, 3, 4, 5, 8

(b) 1, 1, 1, 2, 2, 3, 3, 4, 5, 6, 6

(c) 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 5, 5, 6, 7

(d) 1, 2, 2, 3, 3, 3, 4, 5, 5, 5, 6, 6

The cost is found by adding the numbers given.

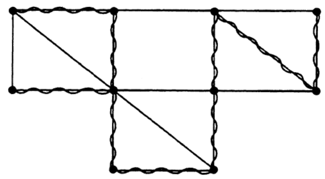
52. (a) If *G* is connected and has 20 vertices, a spanning tree of *G* will have 19 edges.

(b) Any spanning tree of *G* will have 20 vertices.

(c) One can conclude that *G* must have at least 19 edges (when *G* itself is a tree), but nothing more. (If *G* cannot have two edges joining a single pair of vertices, then the largest number of edges G can have is 190.)

53. The spanning tree will have 26 vertices. *H* also will have 26 vertices. The exact number of edges in *H* cannot be determined, but *H* has at least 25 edges and no more edges than the complete graph having 26 vertices.

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



55. Yes.

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

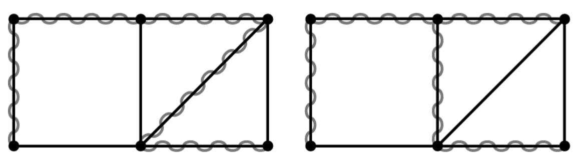
57. Yes. Change all the weights to negative numbers and apply Kruskal's algorithm. The resulting tree works, and the maximum cost is the negative of the answer you get. If the numbers on the edges represent subsidies for using the edges, one might be interested in finding a maximum-cost spanning tree.

58. Air distances may yield a different solution.

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

60. 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.

61. Two different trees with the same cost are shown:



62. If one starts at vertex X_1, the edges are added in the order:

(a) $X_1, X_6, X_2, X_5, X_2, X_2, X_7, X_2, X_3, X_8, X_8, X_9, X_9, X_4$

(b) Starting at *A*, the edges are added in the order *AB*, *BE*, *ED*, *DC*.

63. (a) True

(b) False (unless all the edges of the graph have the same weight)

(c) True

(d) False

(e) False

64. (a) If the vertex costs are neglected, the solution has cost $2+4+8=14$. If the vertex costs are not neglected, using the vertex with weight 5 and the edges at it gives a cost of 24.

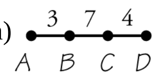
(b) If the vertex costs are neglected, the solution has cost $2+4+7=13$.

65. (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.

66. (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 which 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.

67.

	A	B	C	D
A	0	14	13	5
B	14	0	17	9
C	13	17	0	8
D	5	9	8	0

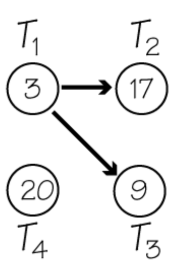
68. (a) The earliest completion time is 37 since the longest path, the unique critical path $T_1 T_4 T_7$ has length 37.

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

69. (a) The earliest completion time is 22 since the longest path, the unique critical path $T_3T_2T_5$, has length 22.

(b) The earliest completion time is 30 since the longest path, the unique critical path $T_3T_5T_7$, has length 30.

70.

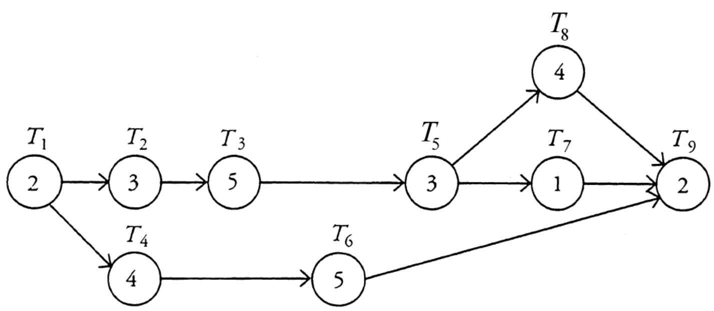


71. The only tasks which if shortened will reduce the earliest completion time are those on the critical path, so in this case, these are the tasks T_1, T_5, and T_7. If T_5 is shortened to 7, then the longest path will have length 28, and this becomes the earliest completion time. The tasks on this critical path are T_1, T_4, and T_7.

72. The critical path is T_1, T_5, T_7 with length 30. 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 28.

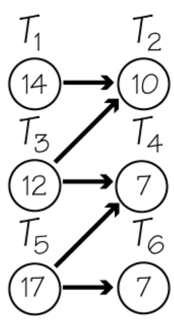
73. Different contractors will have different times and order-requirement digraphs. However, in any sensible order requirement digraph, the laying of the foundation will come before the erection of the side walls and the roof. The fastest time for completing all the tasks will be the length of the longest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

74. One possibility is (times in minutes):



The earliest completion time is 19 minutes.

75. One example is given below.



Word Search Solution

```
U D N K K Z T S B U P F T I U C R I C N A I N O T L I M A H
L E E I U L M M H F E G A H Q R T E R S U M S T P A Z E E E
N P K G O G H G L R O A A M M S A R A W T I O Y O I Y O B R
T R G S A T T M G Z W S C R S F I L J F D N S B I S O U M I
R R D M O C I D L Q F T L V O G S A E T E I G A O O M O M S
A T G L S K R U S K A L S A L G O R I T H M R I Z L X O V L
V M N O K V O I E U T S R S G O P I D I S U T E O O A S M B
E H K E N M G R A S O R E Y K A O Q E K N M H F T R T E D P
L A N Y O E L I D F T O E D S S N E N P L C F M I E U C E A
I F M F T K A E E J E B D O H T E M E C R O F E T U R B S A
N V S E X C R I T I C A L P A T H F P C S S C P L Z N Z R V
G N I T N U O C F O E L P I C N I R P L A T N E M A D N U F
S P M Q Q R B R L E S E Z X A T S L Y S Z H H V D I I E H R
A R R N B E H P A R G E T E L P M O C F A A M T D S C N E T
L S G P T H G I E W E R E H F X S J O M A M Q S F S R I U M
E T R H H D I S E F I G F O G Q D E S F S I F T E K R E R N
S O R T E D E D G E S A L G O R I T H M R L I S N O V L Q Y
M E U E E T N L B J C S P T C B P E I E P T I A H B C E K J
A B I N E Q T R Z M C M H T I R O G L A O O E K O N M A W R
N C E B A F S X W H S E C P Y A G H I Y I N W S Z C F L F S
P L A G R E E D Y A L G O R I T H M E Z T I M Y L S H G F I
R L G H E U R I S T I C A L G O R I T H M A I C N S S O R M
O I X H R R A S E E R T F O D O H T E M E N V E I J S R O F
B O S R M R E E R T G N I N N A P S T S O C M U M I N I M C
L N U B E E N E B D Q Q F H N D L C J E S I O R S B X T N L
E R E I A Z F Y L Q Y P D Z C E D F E W S R B O P E E H E G
M D G L S I S M E L B O R P E T E L P M O C P N B M Z M L R
R X M U T E C H P A R G I D T N E M E R I U Q E R R E D R O
H I L E Y M J H N W O K S B K P P X L M F I T O H Y S C U E
N D F R P P M F E G A F O T T N G R E E R T G N I N N A P S
```