Class Work 18

quality web pages, called **seeds**, and explore all links on these pages. As the web crawler proceeds, it adds the urls of all links on the new pages it visits, adding them to the **crawl frontier**. These urls are ordered according to a particular policy, determining the order in which websites are explored.

When seeds have been selected well, Google's approach has been found to quickly reach high quality pages that have many links pointing to them. However, the quality of pages reached decreases as the web crawl continues. This reaches popular web pages well, but it does not reach many other useful, but less popular, web pages. If seed pages do not yield good results, DFS can be used to find candidates for high quality pages, starting at the seeds or other web pages. Also, DFS can be used to reach parts of the web not reached by BFS when it is restricted to a particular number of levels. ◀

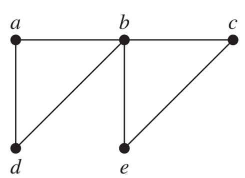
Exercises

WHOLE PAGE. (1→12)

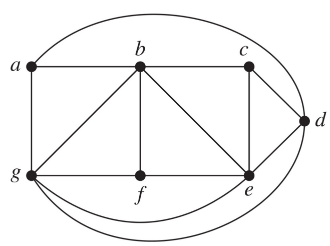
1. How many edges must be removed from a connected graph with n vertices and m edges to produce a spanning tree?

In Exercises 2–6 find a spanning tree for the graph shown by removing edges in simple circuits.

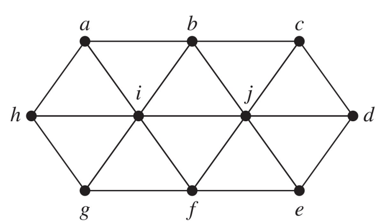
2.



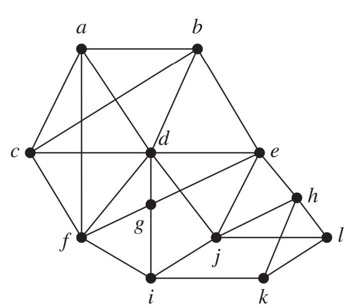
3.



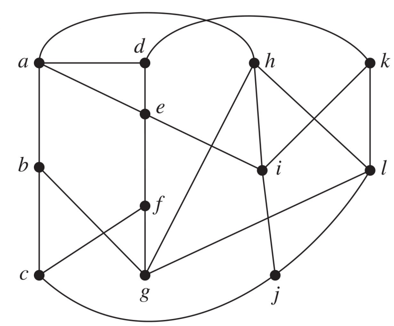
4.



5.



6.

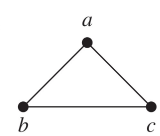


7. Find a spanning tree for each of these graphs.

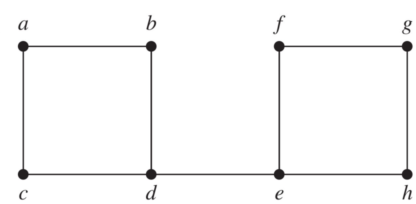
a) K_5 **b)** $K_{4,4}$ **c)** $K_{1,6}$
d) Q_3 **e)** C_5 **f)** W_5

In Exercises 8–10 draw all the spanning trees of the given simple graphs.

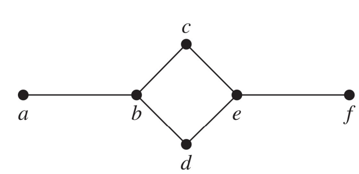
8.



9.



10.



∗11. How many different spanning trees does each of these simple graphs have?

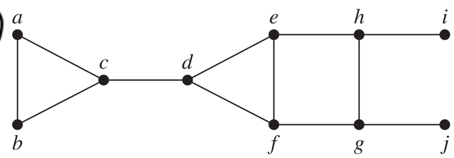
a) K_3 **b)** K_4 **c)** $K_{2,2}$ **d)** C_5

∗12. How many nonisomorphic spanning trees does each of these simple graphs have?

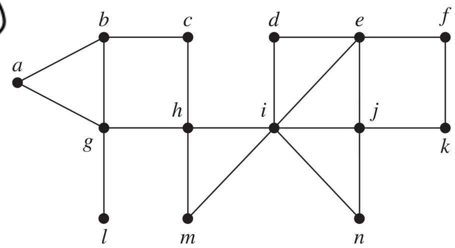
a) K_3 **b)** K_4 **c)** K_5

In Exercises 13–15 use depth-first search to produce a spanning tree for the given simple graph. Choose a as the root of this spanning tree and assume that the vertices are ordered alphabetically.

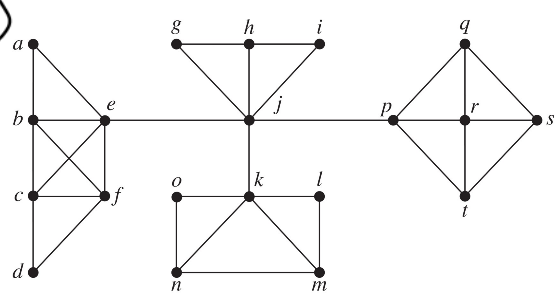
13.



14.



15.



16. Use breadth-first search to produce a spanning tree for each of the simple graphs in Exercises 13–15. Choose a as the root of each spanning tree.

17. Use depth-first search to find a spanning tree of each of these graphs.

a) W_6 (see Example 7 of Section 10.2), starting at the vertex of degree 6

b) K_5

c) $K_{3,4}$, starting at a vertex of degree 3

~~**d)** Q_3~~

18. Use breadth-first search to find a spanning tree of each of the graphs in Exercise 17.

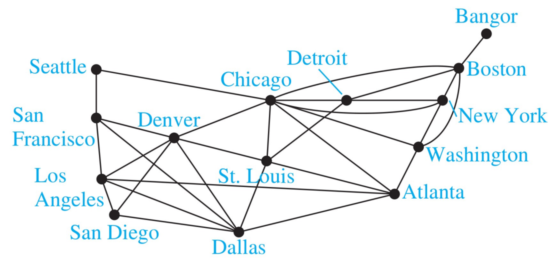
19. Describe the trees produced by breadth-first search and depth-first search of the wheel graph W_n, starting at the vertex of degree n, where n is an integer with $n \geq 3$. (See Example 7 of Section 10.2.) Justify your answers.

20. Describe the trees produced by breadth-first search and depth-first search of the complete graph K_n, where n is a positive integer. Justify your answers.

21. Describe the trees produced by breadth-first search and depth-first search of the complete bipartite graph $K_{m,n}$, starting at a vertex of degree m, where m and n are positive integers. Justify your answers.

22. Describe the tree produced by breadth-first search and depth-first search for the n-cube graph Q_n, where n is a positive integer.

23. Suppose that an airline must reduce its flight schedule to save money. If its original routes are as illustrated here, which flights can be discontinued to retain service between all pairs of cities (where it may be necessary to combine flights to fly from one city to another)?



24. Explain how breadth-first search or depth-first search can be used to order the vertices of a connected graph.

***25.** Show that the length of the shortest path between vertices v and u in a connected simple graph equals the level number of u in the breadth-first spanning tree of G with root v.

26. Use backtracking to try to find a coloring of each of the graphs in Exercises 7–9 of Section 10.8 using three colors.

27. Use backtracking to solve the n-queens problem for these values of n.

a) $n = 3$ **b)** $n = 5$ **c)** $n = 6$

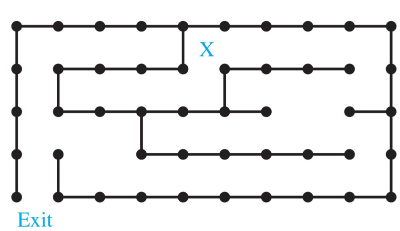
28. Use backtracking to find a subset, if it exists, of the set $\{27, 24, 19, 14, 11, 8\}$ with sum

a) 20. **b)** 41. **c)** 60.

29. Explain how backtracking can be used to find a Hamilton path or circuit in a graph.

30. **a)** Explain how backtracking can be used to find the way out of a maze, given a starting position and the exit position. Consider the maze divided into positions, where at each position the set of available moves includes one to four possibilities (up, down, right, left).

b) Find a path from the starting position marked by X to the exit in this maze.



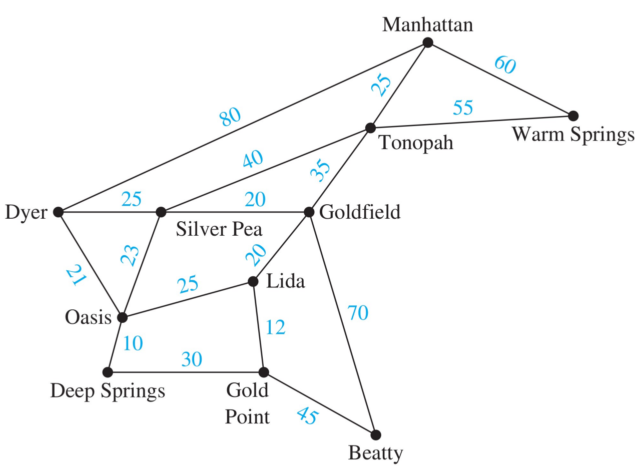
edge in the simple circuit that does not belong to S_{k+1} because S_{k+1} is a tree. By starting at an endpoint of e_{k+1} that is also an endpoint of one of the edges $e_1, \ldots, e_k$, and following the circuit until it reaches an edge not in S_{k+1}, we can find an edge e not in S_{k+1} that has an endpoint that is also an endpoint of one of the edges $e_1, e_2, \ldots, e_k$.

By deleting e from T and adding e_{k+1}, we obtain a tree T' with $n-1$ edges (it is a tree because it has no simple circuits). Note that the tree T' contains $e_1, e_2, \ldots, e_k, e_{k+1}$. Furthermore, because e_{k+1} was chosen by Prim's algorithm at the $(k+1)$st step, and e was also available at that step, the weight of e_{k+1} is less than or equal to the weight of e. From this observation, it follows that T' is also a minimum spanning tree, because the sum of the weights of its edges does not exceed the sum of the weights of the edges of T. This contradicts the choice of k as the maximum integer such that a minimum spanning tree exists containing $e_1, \ldots, e_k$. Hence, $k = n-1$, and $S = T$. It follows that Prim's algorithm produces a minimum spanning tree. ◁

It can be shown (see [CoLeRiSt09]) that to find a minimum spanning tree of a graph with m edges and n vertices, Kruskal's algorithm can be carried out using $O(m \log m)$ operations and Prim's algorithm can be carried out using $O(m \log n)$ operations. Consequently, it is preferable to use Kruskal's algorithm for graphs that are **sparse**, that is, where m is very small compared to $C(n, 2) = n(n-1)/2$, the total number of possible edges in an undirected graph with n vertices. Otherwise, there is little difference in the complexity of these two algorithms.

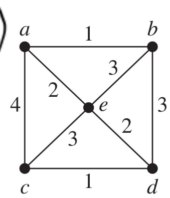
Exercises

1. The roads represented by this graph are all unpaved. The lengths of the roads between pairs of towns are represented by edge weights. Which roads should be paved so that there is a path of paved roads between each pair of towns so that a minimum road length is paved? (*Note:* These towns are in Nevada.)

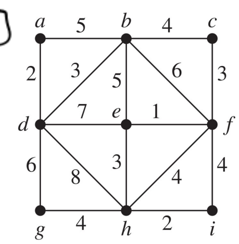


In Exercises 2–4 use Prim's algorithm to find a minimum spanning tree for the given weighted graph.

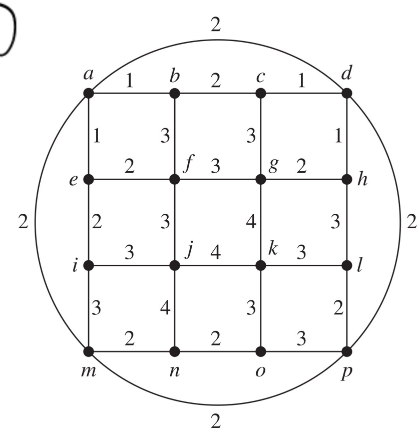
2.



3.



4.



5. Use Kruskal's algorithm to design the communications network described at the beginning of the section.

6. Use Kruskal's algorithm to find a minimum spanning tree for the weighted graph in Exercise 2.

7. Use Kruskal's algorithm to find a minimum spanning tree for the weighted graph in Exercise 3.

8. Use Kruskal's algorithm to find a minimum spanning tree for the weighted graph in Exercise 4.