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.

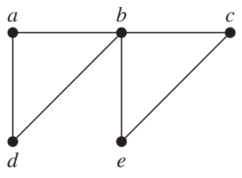
When seeds have been selected well, Google's approach has been found to quickly reach high quality pages that have many links pointing to them. However, the quality of pages reached decreases as the web crawl continues. This reaches popular web pages well, but it does not reach many other useful, but 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

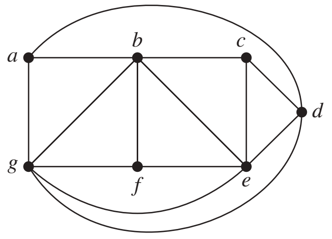
1. How many edges must be removed from a connected graph with n vertices and m edges to produce a spanning tree?

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

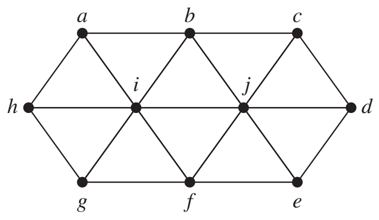
2.



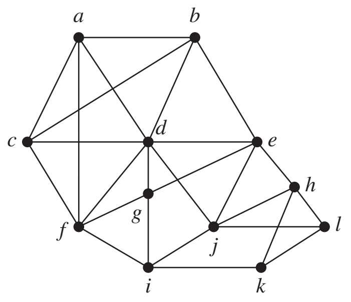
3.



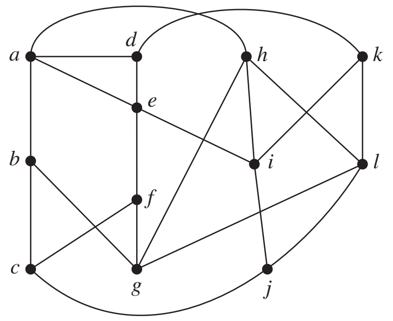
4.



5.



6.



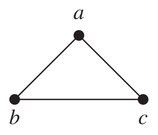
7. Find a spanning tree for each of these graphs.

a) K_5 **b)** $K_{4,4}$ **c)** $K_{1,6}$

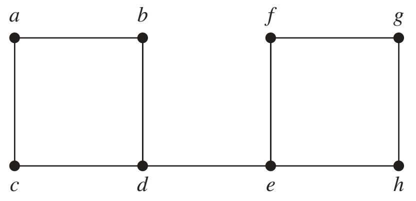
d) Q_3 **e)** C_5 **f)** W_5

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

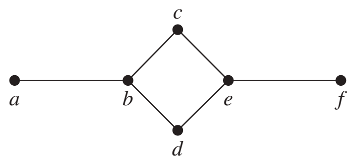
8.



9.



10.



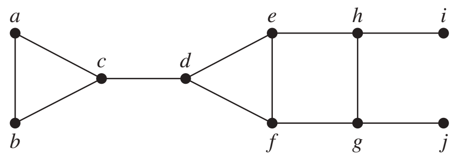
***11.** How many different spanning trees does each of these simple graphs have?

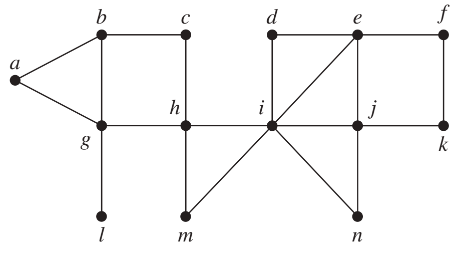
a) K_3 **b)** K_4 **c)** $K_{2,2}$ **d)** C_5

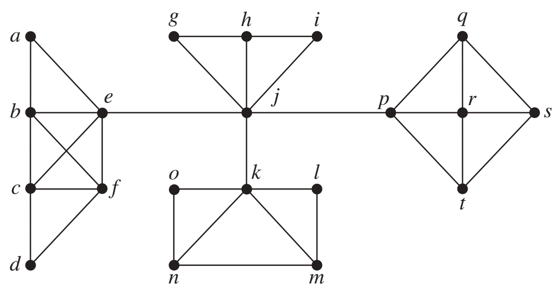
***12.** How many nonisomorphic spanning trees does each of these simple graphs have?

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.

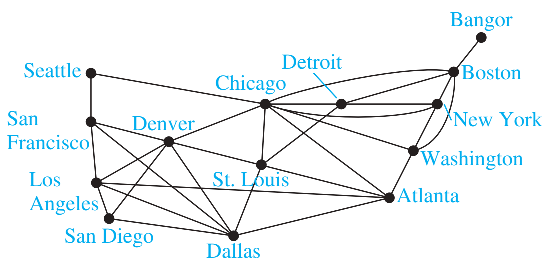
19. Describe the trees produced by breadth-first search and depth-first search of the wheel graph W_n, starting at the vertex of degree n, where n is an integer with $n \geq 3$. (See Example 7 of Section 10.2.) Justify your answers.

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)** $n = 6$

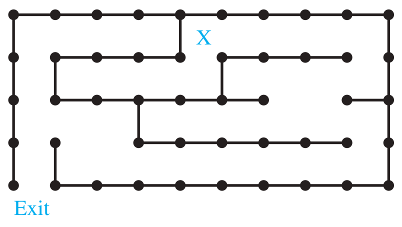
28. Use backtracking to find a subset, if it exists, of the set $\{27, 24, 19, 14, 11, 8\}$ with sum

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.



A **spanning forest** of a graph G is a forest that contains every vertex of G such that two vertices are in the same tree of the forest when there is a path in G between these two vertices.

31. Show that every finite simple graph has a spanning forest.

32. How many trees are in the spanning forest of a graph?

33. How many edges must be removed to produce the spanning forest of a graph with n vertices, m edges, and c connected components?

34. Let G be a connected graph. Show that if T is a spanning tree of G constructed using breadth-first search, then an edge of G not in T must connect vertices at the same level or at levels that differ by 1 in this spanning tree.

35. Explain how to use breadth-first search to find the length of a shortest path between two vertices in an undirected graph.

36. Devise an algorithm based on breadth-first search that determines whether a graph has a simple circuit, and if so, finds one.

37. Devise an algorithm based on breadth-first search for finding the connected components of a graph.

38. Explain how breadth-first search and how depth-first search can be used to determine whether a graph is bipartite.

39. Which connected simple graphs have exactly one spanning tree?

40. Devise an algorithm for constructing the spanning forest of a graph based on deleting edges that form simple circuits.

41. Devise an algorithm for constructing the spanning forest of a graph based on depth-first searching.

42. Devise an algorithm for constructing the spanning forest of a graph based on breadth-first searching.

43. Let G be a connected graph. Show that if T is a spanning tree of G constructed using depth-first search, then an edge of G not in T must be a back edge, that is, it must connect a vertex to one of its ancestors or one of its descendants in T.

44. When must an edge of a connected simple graph be in every spanning tree for this graph?

45. For which graphs do depth-first search and breadth-first search produce identical spanning trees no matter which vertex is selected as the root of the tree? Justify your answer.

46. Use Exercise 43 to prove that if G is a connected, simple graph with n vertices and G does not contain a simple path of length k then it contains at most $(k-1)n$ edges.

47. Use mathematical induction to prove that breadth-first search visits vertices in order of their level in the resulting spanning tree.

48. Use pseudocode to describe a variation of depth-first search that assigns the integer n to the nth vertex visited in the search. Show that this numbering corresponds to the numbering of the vertices created by a preorder traversal of the spanning tree.

49. Use pseudocode to describe a variation of breadth-first search that assigns the integer m to the mth vertex visited in the search.

***50.** Suppose that G is a directed graph and T is a spanning tree constructed using breadth-first search. Show that every edge of G has endpoints that are at the same level or one level higher or lower.

51. Show that if G is a directed graph and T is a spanning tree constructed using depth-first search, then every edge not in the spanning tree is a **forward edge** connecting an ancestor to a descendant, a **back edge** connecting a descendant to an ancestor, or a **cross edge** connecting a vertex to a vertex in a previously visited subtree.

***52.** Describe a variation of depth-first search that assigns the smallest available positive integer to a vertex when the algorithm is totally finished with this vertex. Show that in this numbering, each vertex has a larger number than its children and that the children have increasing numbers from left to right.

Let T_1 and T_2 be spanning trees of a graph. The **distance** between T_1 and T_2 is the number of edges in T_1 and T_2 that are not common to T_1 and T_2.

53. Find the distance between each pair of spanning trees shown in Figures 3(c) and 4 of the graph G shown in Figure 2.

***54.** Suppose that T_1, T_2, and T_3 are spanning trees of the simple graph G. Show that the distance between T_1 and T_3 does not exceed the sum of the distance between T_1 and T_2 and the distance between T_2 and T_3.

****55.** Suppose that T_1 and T_2 are spanning trees of a simple graph G. Moreover, suppose that e_1 is an edge in T_1 that is not in T_2. Show that there is an edge e_2 in T_2 that is not in T_1 such that T_1 remains a spanning tree if e_1 is removed from it and e_2 is added to it, and T_2 remains a spanning tree if e_2 is removed from it and e_1 is added to it.

***56.** Show that it is possible to find a sequence of spanning trees leading from any spanning tree to any other by successively removing one edge and adding another.

A **rooted spanning tree** of a directed graph is a rooted tree containing edges of the graph such that every vertex of the graph is an endpoint of one of the edges in the tree.

57. For each of the directed graphs in Exercises 18–23 of Section 10.5 either find a rooted spanning tree of the graph or determine that no such tree exists.

***58.** Show that a connected directed graph in which each vertex has the same in-degree and out-degree has a rooted spanning tree. [*Hint:* Use an Euler circuit.]

***59.** Give an algorithm to build a rooted spanning tree for connected directed graphs in which each vertex has the same in-degree and out-degree.

***60.** Show that if G is a directed graph and T is a spanning tree constructed using depth-first search, then G contains a circuit if and only if G contains a back edge (see Exercise 51) relative to the spanning tree T.

***61.** Use Exercise 60 to construct an algorithm for determining whether a directed graph contains a circuit.