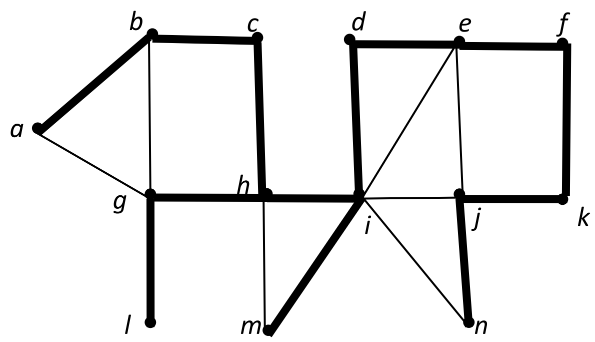
Sample Solutions on Sec.11.4-11.5

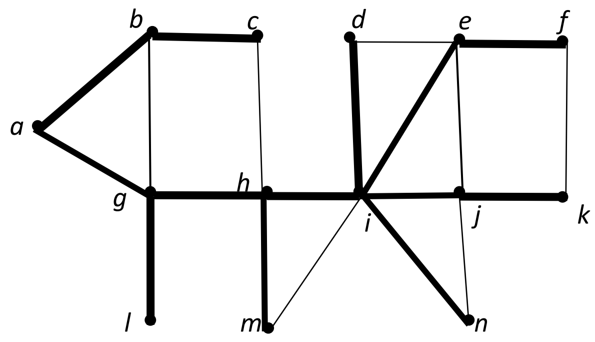
Sec. 11.4 4, 14, 16(14), 29

4 We can remove these edges to produce a spanning tree: $\{a,i\}$, $\{b,i\}$, $\{b,j\}$, $\{c,d\}$, $\{c,j\}$, $\{d,e\}$, $\{e,j\}$, $\{f,i\}$, $\{f,j\}$, and $\{g,i\}$. There are many other possible answers, corresponding to different choices of edges to remove.

14 The tree is shown in heavy lines. It is produced by starting at *a* and continuing as far as possible without backtracking, choosing the first unused vertex (in alphabetical order) at each point. When the path reaches vertex *l*, we need to backtrack. Backtracking to *h*, we can then form the path all the way to *n* without further backtracking. Finally we backtrack to vertex *i* to pick up vertex *m*.



16(14) If we start at vertex *a* and **use alphabetical order,** then the breadth-first search spanning tree is **unique**. We first fan out from vertex *a*, picking up the edges $\{a,b\}$ and $\{a,g\}$. Then we fan out from *b* to get edges $\{b,c\}$ and $\{b,g\}$. Next we fan out from *g* to get edges $\{g,l\}$ and $\{g,h\}$. This process continues until we have the entire tree shown in heavy lines below.



29 Assume that the graph has vertices $v_1, v_2, \ldots, v_n$. In looking for a Hamilton circuit we may as well start building a path at v_1. The general step is as follows. We extend the path if we can, to a new vertex (or to v_1 if this will complete the Hamilton circuit) adjacent to the vertex we are at. If we cannot extend the path any further, then we backtrack to the last previous vertex in the path and try other untried extensions from that vertex. The procedure for Hamilton paths is the same, except that we have try all possible starting vertices, and we do not allow a return to the starting vertex, stopping instead when we have a path of the right length.

Sec. 11.5 3,7,12

3 We start with the minimum weight edge $\{e,f\}$. The least weight edges incident to the tree constructed so far are edges $\{c,f\}$ and $\{e,h\}$, each with weight 3, so we add one of them to the tree (we will break ties using alphabetical order, so we add $\{c,f\}$). Next we add edge $\{e,h\}$, and then edge$\{h,i\}$, which has a smaller weight but has just become eligible for addition. The edges continue to be added in the following order (note that ties are broken using alphabetical order): $\{b,c\}$, $\{b,d\}$, $\{a,d\}$, and $\{g,h\}$. The total weight of the minimum spanning tree is 22.

7 The edges are added in the following order (with Kruskal's algorithm, we add at each step the shortest edge that will not complete a simple circuit): $\{e,f\}$, $\{a,d\}$, $\{h,i\}$, $\{b,d\}$, $\{c,f\}$, $\{e,h\}$, $\{b,c\}$, and $\{g,h\}$. The total weight of the minimum spanning tree is 22.

12 If we simply replace the word "smallest" with the word "largest" (and replace the word "minimum" in the comment with the word "maximum") in Algorithm 2, then the resulting algorithm will find a maximum spanning tree.