CHAPTER 9
TREES

SECTION 9.1 Basic Definitions for Trees

1. We know that a tree with n vertices has $n-1$ edges. We also know that the sum of the degrees must equal twice the number of edges. Therefore the sum of the degrees must be $2(n-1)$. Thus the average is $2(n-1)/n = 2-(2/n) < 2$.

3. **(a)** The number of edges in a forest is equal to the number of vertices minus the number of components (a tree, with just one component, has $v-1$ edges, and we need to remove one more edge to create each new component). Therefore there are $72-7=65$ edges in this forest.

   (b) We know that $l=(m-1)i+1$. Therefore $l=2\cdot 7+1=15$.

   (c) Plugging $m=5$ and $l=101$ into the equation $l=(m-1)i+1$ yields $101=4i+1$, whence $i=25$.

   (d) We know that $n \le 2^{h+1}-1$, with equality holding if the tree is full and all the leaves are at level h. Thus the maximum possible value for n is $2^{5+1}-1=63$.

5. **(a)** f, g

   (b) none (m is a leaf)

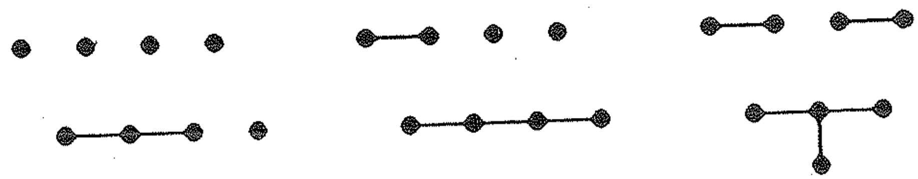
   (c) h, i, j (the children of the children of b)

   (d) l, m, n (the other children of k's parent, f)

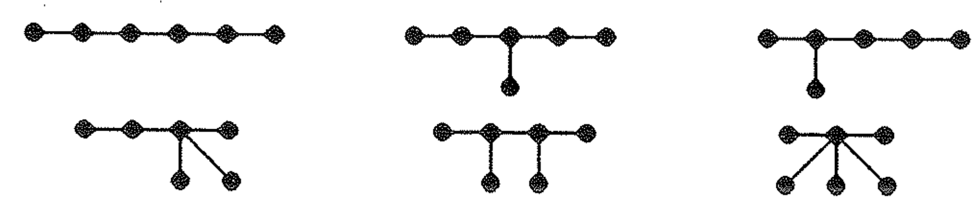
   (e) f, g, k, l, m, n (all the vertices in the subtree rooted at d, other than d)

   (f) a, b, e, j (the vertices on the path from o back to the root, other than o)

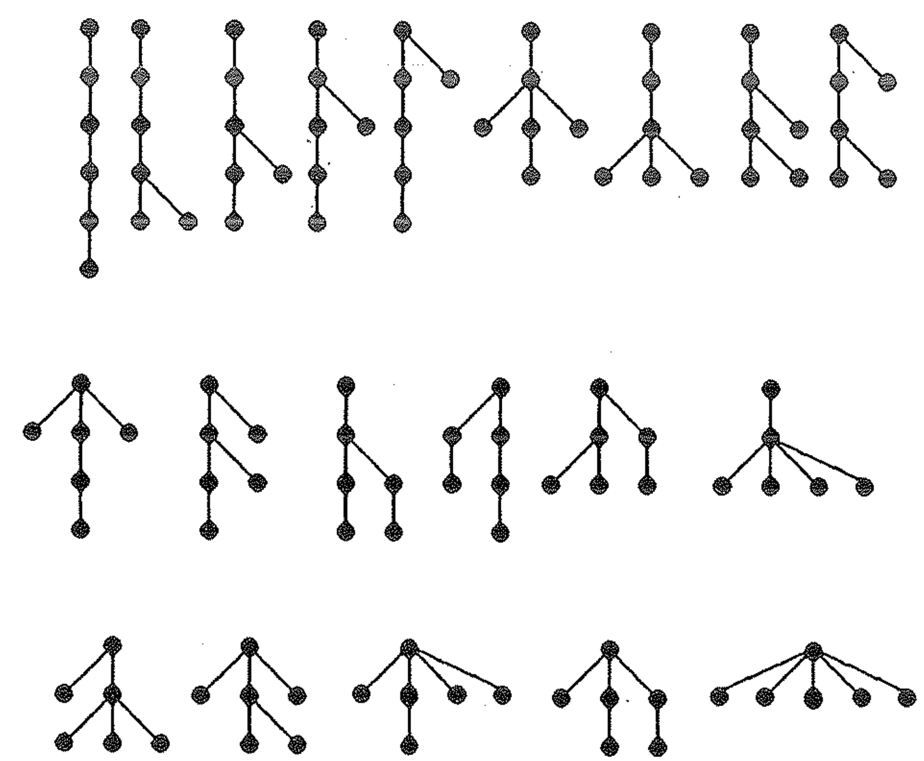
7. **(a)** There are six such forests, as shown here. There is only one forest for each of the cases in which there are four components, three components, two components of equal size, and two components of unequal size. The other cases are when the forest is a tree, and there are clearly only the two possibilities shown.



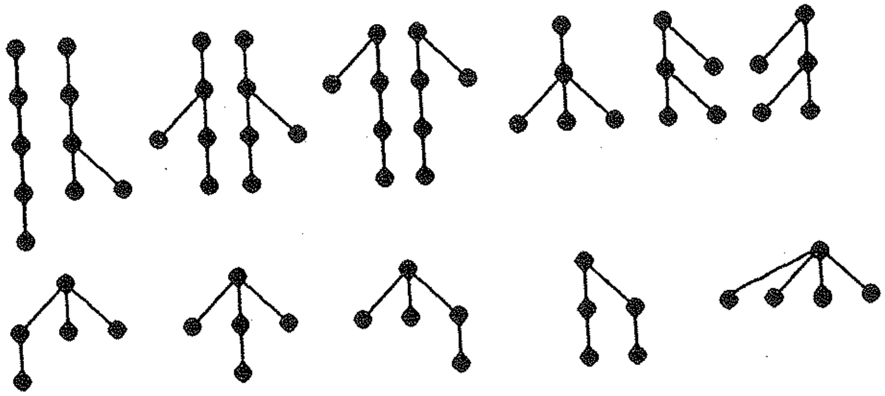
(b) We organize the search by concentrating on the longest path. There is only one tree in which all six vertices are in a path. If only five vertices are in a path, then the sixth vertex can be attached either to the middle vertex or to a vertex neither in the middle nor on the end. If only four vertices are in a path, then the other two vertices can be attached either to the same central vertex or two different central vertices, as shown. Finally, there may be no path of more than three vertices, in which case the tree must look as in the final picture here.



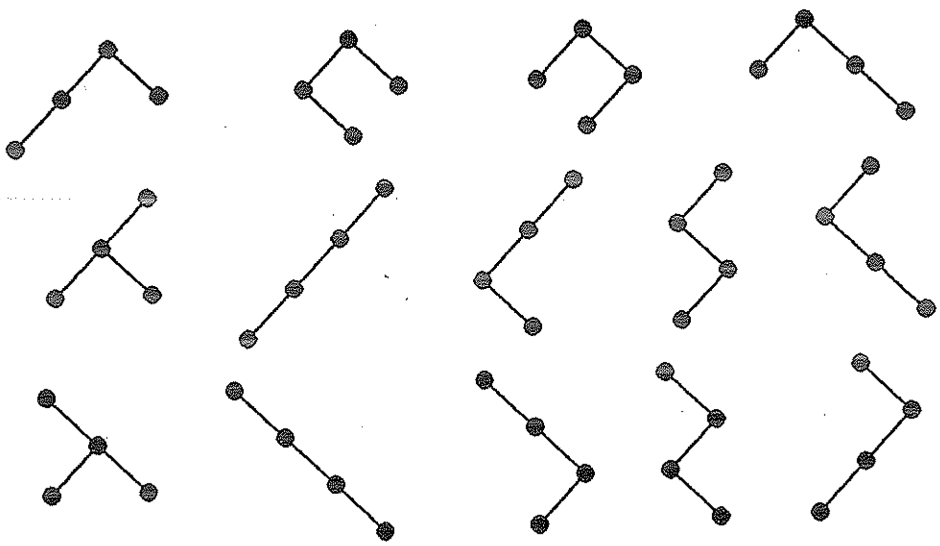
(c) Again we concentrate on the length of the longest path—in this case the height of the tree. The following pictures show the 20 possibilities.



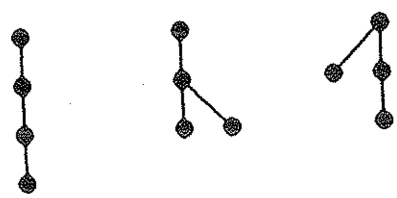
(d) Again we concentrate on the length of the longest path (i.e., the height of the tree). We must be careful to distinguish between different orders of subtrees, however. The following pictures show the 14 possibilities.



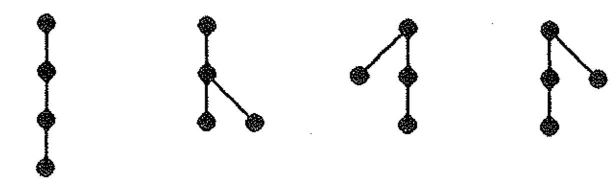
(e) We need to distinguish between left children and right children throughout. There are 14 binary trees with four vertices.



(f) Of the 14 trees found in part (e), only three are distinguishable as rooted trees, as shown here.



(g) Of the three trees found in part (f), the one on the far right has two different incarnations as a rooted tree, since either the leaf of the nonleaf can be the first child of the root.



9. **(a)** Let G be a connected graph with n vertices and $n-1$ edges. Suppose (for an indirect proof) that G is still connected when some edge is removed. If this smaller graph has no closed trails, then it is by definition a tree. If it has a closed trail, then we can delete any edge in this closed trail without disconnecting the graph, giving us a smaller connected graph. We continue in this manner, deleting edges in closed trails, until we obtain a tree. But by $1 \to 2$, this tree must have $n-1$ edges, a contradiction to the fact that we deleted an edge at the first step.

(b) Let G be a graph with no closed trails such that if any edge not in G is added to G, then the resulting graph contains a cycle. Suppose (for an indirect proof) that G is not a tree; thus G is not connected. Let u and v be in different components of G. Let G' be G with edge uv added. By the hypothesis, G' contains a cycle C, necessarily using edge uv. But then C without uv must be a path connecting u and v in G, contradicting the choice of u and v. Therefore G is connected.

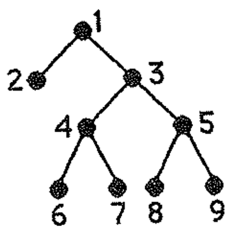
11. **(a)** We can consider any tree as a rooted tree, by specifying one vertex to be the root. Let one part consist of all vertices at even-numbered levels; and let the other part consist of all vertices at odd-numbered levels. Since edges only join vertices at adjacent levels, every edge in the tree joins vertices in different parts. Thus the tree is bipartite.

(b) Clearly $K_{1,n}$ and $K_{m,1}$ are trees—they are called stars. If both m and n are greater than 1, however, then there is a cycle: u_1, v_1, u_2, v_2, u_1 (with the obvious notation). Thus the necessary and sufficient conditions are that $m=1$ or $n=1$. An alternative proof goes along the following lines. The number of edges in $K_{m,n}$ is mn, and the number of vertices is $m+n$. Since $K_{m,n}$ is connected, Theorem 1 tells us that it is a tree if and only if $mn = m+n-1$. This is equivalent to $mn-m-n+1=0$, which factors as $(m-1)(n-1)=0$. Clearly this equality holds if and only if $m=1$ or $n=1$.

13. The height of a tree is the maximum length of a path from the root to a leaf. Such a path must consist of an edge from the root to a child v_i of the root, followed by a path from this child to a leaf in its subtree T_i. Thus the longest path has length one greater than the maximum of the heights of its immediate subtrees T_i.

15. By Theorem 5 (with $m=2$), $l \le 2^h$. Thus $\log l \le h$, so $h \ge \lceil \log l \rceil$ (since h is an integer). Also by Theorem 5, $n \le 2^{h+1}-1$, so $h+1 \ge \lceil \log(n+1) \rceil$, or $h \ge \lceil \log(n+1) \rceil - 1$. For any $m \ge 2$ essentially the same analysis applies using logarithms base m. Note that asymptotically the two results are the same up to a constant factor, since $O(\lceil \log_m (m-1)n+1 \rceil - 1) = O(\lceil \log(n+1) \rceil - 1)$.

17. One such tree is shown below. In the order the vertices are numbered, each vertex that will end up as an internal vertex sprouts its two children. The tree begins with vertex 1. It grows two children, so now the tree contains vertices 1, 2, and 3. Vertex 2 has no children, but vertex 3 then grows its two children, giving us the tree extending down to level 2; and so on.



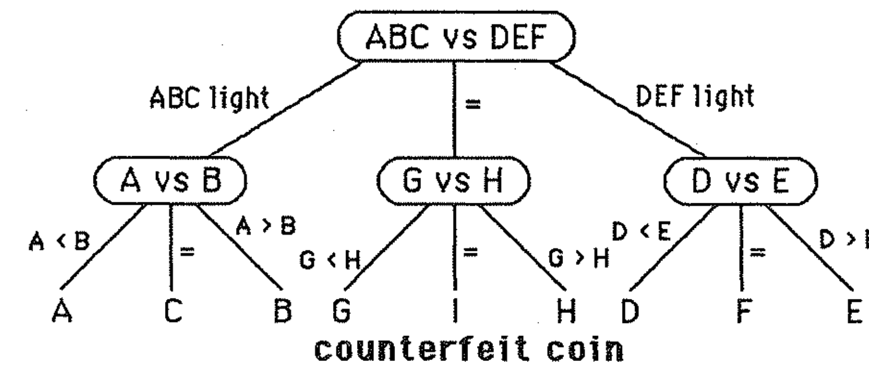
19. There are $h-l$ levels below level l in the tree. Therefore the subtree rooted at the given vertex v at level l has height at most $h-l$. Thus by Theorem 5 it has at most $2^{h-l+1}-1$ vertices. All of them except v itself are descendants of v, so the answer is $2^{h-l+1}-2$.

21. It is routine algebra to solve two equations simultaneously for the two unknowns that we are interested in, treating the other two variables as constants. In terms of i and l, we have $m=[(l-1)/i]+1$ and $n=i+l$. In terms of i and m, we have $l=(m-1)i+1$ and $n=mi+1$. In terms of i and n, we have $l=n-i$ and $m=(n-1)/i$. In terms of l and m, we have $i=(l-1)/(m-1)$ and $n=[(l-1)/(m-1)]+l$. In terms of l and n, we have $i=n-l$ and $m=(n-1)/(n-l)$. Finally, in terms of m and n, we have $i=(n-1)/m$ and $l=n-[(n-1)/m]$.

23. **(a)** Any weighing can be modeled as an internal vertex of the tree, with 3 children, corresponding to the possibilities that the left pan is light, the right pan is light, or the pans balance. Leaves are "solutions"—decisions as to which coin is under weight. It is a full 3-ary tree.

(b) There are nine outcomes, since any of the nine coins can be underweight. A 3-ary tree of height 2 can have nine leaves, so we can hope to get by with a tree of height 2—in other words, with two weighings.

(c) Let us label the coins with the letters A through I. Our first weighing involves two groups of three coins, say A, B, and C in one pan weighed against D, E, and F in the other. If the scale balances, then we know that the counterfeit is among G, H, and I. Otherwise we know which of the first two groups the light-weight coin is in. In each case, we have narrowed the choice to three coins. We then weigh two of these coins against each other, and it is clear that after this second weighing we have identified the counterfeit. We can show this procedure with the following tree. The leaves list the counterfeit coin found.



25. These definitions are really the same ones that are used in common speech about relatives.

(a) v's parent is u's sibling

(b) u's parent and v's parent are siblings

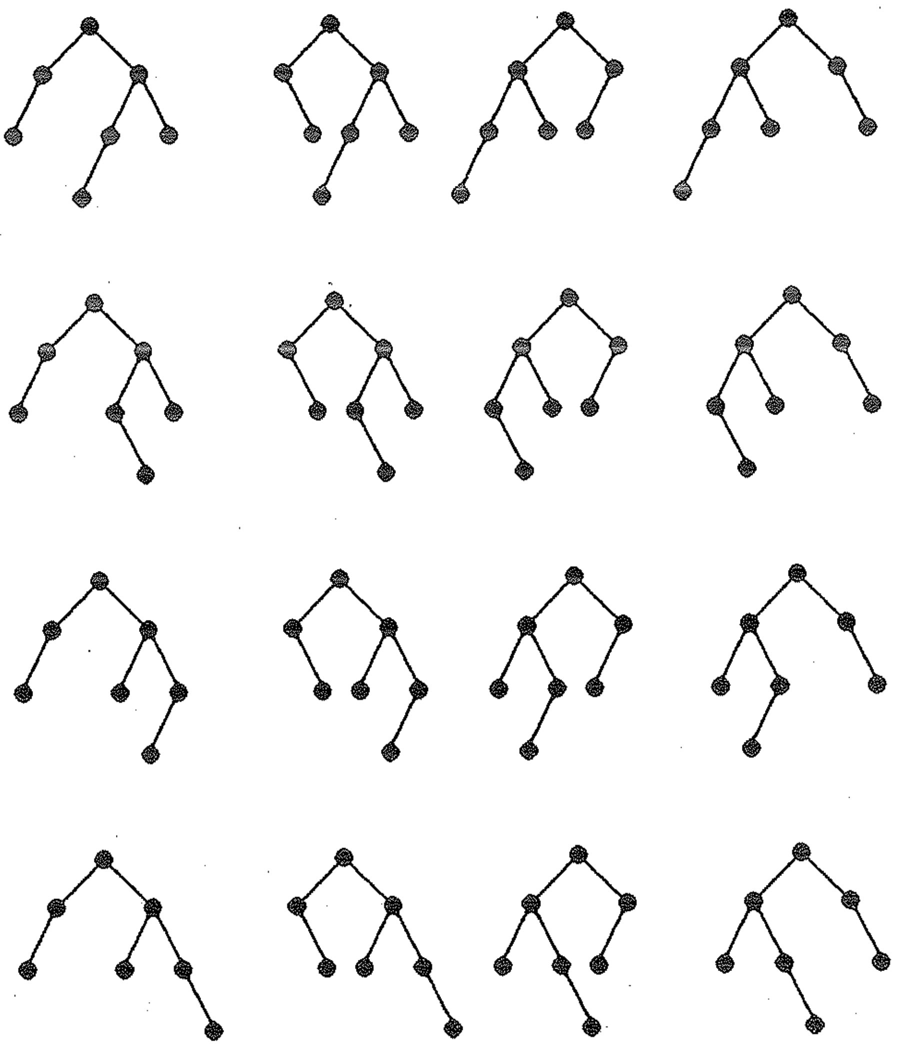
(c) We need to find a common ancestor $n+1$ generations back but no common ancestor n generations back. This means that there exist vertices u_i and v_i for $i = 1, 2, \ldots, n$ and vertex w (the common ancestor) such that both w, u_1, u_2, ..., u_n, u and w, v_1, v_2, ..., v_n, v are directed paths in the tree and $u_1 \neq v_1$.

(d) This is similar to part (c), except that the path must be longer (by r levels) to one of the vertices than to the other. Thus either there exist vertices u_i for $i = 1, 2, \ldots, n$, vertices v_i for $i = 1, 2, \ldots, n+r$, and vertex w such that both w, u_1, u_2, ..., u_n, u and w, v_1, v_2, ..., v_{n+r}, v are directed paths in the tree and $u_1 \neq v_1$, or else there exist vertices u_i for $i = 1, 2, \ldots, n+r$, vertices v_i for $i = 1, 2, \ldots, n$, and vertex w such that both w, u_1, u_2, ..., u_{n+r}, u and w, v_1, v_2, ..., v_n, v are directed paths in the tree and $u_1 \neq v_1$.

27. Make T a rooted tree from any vertex w. Let v be a vertex at maximum level in this rooted tree. Now redraw the tree rooted at v. The height of this rooted tree is the length of the longest path in the tree; in other words, a longest path in the tree runs from v to some leaf at the bottom level in this second rooted tree. In order to see that this is so, we must prove that v is an end of a longest path. Suppose not. Let P be some path longer than any path with v as endpoint, let u_1 be one of its endpoints, let u_3 be the other endpoint, and let u_2 be the point on P nearest the root w of our first rooted version of this tree. If there is a point u on the portion of P from u_1 to u_2 that is an ancestor of v, then replacing the portion of P from u to u_1 by the path from u to v makes a path at least as long as P (since v is at a bottom-most level), a contradiction. Otherwise, the path from v to u_1 is at least as long as P, another contradiction.

29. **(a)** The trees in F_2 must have either the unique tree in F_0 as their right subtree and one of the two trees in F_1 as their left subtree, as shown in the first two pictures here, or vice versa, as shown in the last two pictures.

There are four trees to choose from for the immediate subtrees of an element of F_3, namely any of the elements of F_2 just pictured. For each of these, there are two choices for the other immediate subtree (an element of F_2) and two choices as to which side each tree goes on. This gives us the sixteen trees shown here.



(b) We proceed by induction. The statement is clearly true for F_0 and F_1. Assume the inductive hypothesis. If $T \in F_n$ is formed with $T_1 \in F_{n-1}$ and $T_2 \in F_{n-2}$ as its immediate subtrees, then $\text{height}(T) = 1+\max(\text{height}(T_1), \text{height}(T_2)) = 1+(n-1) = n$.

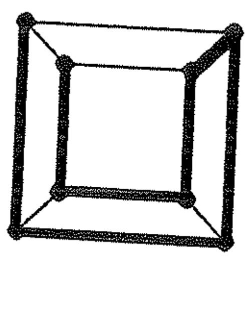
(c) We derive the recurrence relation as we reasoned in constructing F_3 in part (a). There are a_{n-1} choices for the element of F_{n-1} to be one of the immediate subtrees and a_{n-2} choices for the element of F_{n-2} to be the other immediate subtree. There are two choices as to which subtree goes on the left. Therefore by the multiplication principle we have $a_n = 2a_{n-1}a_{n-2}$. The initial conditions are given in the statement of the problem: $a_0 = 1$ and $a_1 = 2$. Thus we can compute, for example, that $a_2 = 2\cdot 2\cdot 1 = 4$, $a_3 = 2\cdot 4\cdot 2 = 16$ (both of which we saw in part (a)), $a_4 = 2\cdot 16\cdot 4 = 128$, and so on.

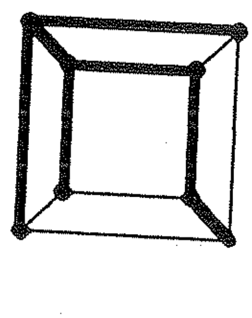
(d) This recurrence relation is not linear, but it can be turned into a linear one by taking logarithms, since then the multiplications will be turned into additions. Specifically, let $b_n = \log a_n$. Applying log to the recurrence relation yields $\log a_n = \log 2 + \log a_{n-1} + \log a_{n-2}$, which says simply $b_n = b_{n-1} + b_{n-2} + 1$. The initial conditions become $b_0 = \log a_0 = \log 1 = 0$ and $b_1 = \log a_1 = \log 2 = 1$. We can apply the techniques of Section 7.3, but let us try something simpler. If we compute several terms from the recurrence, we obtain $b_2 = 2$, $b_3 = 4$, $b_4 = 7$, $b_5 = 12$, $b_6 = 20$, $b_7 = 33$, and so on. These numbers look very similar to the terms of the Fibonacci sequence, which starts out $f_0 = 1$, $f_1 = 1$, $f_2 = 2$, $f_3 = 3$, $f_4 = 5$, $f_5 = 8$, $f_6 = 13$, $f_7 = 21$, $f_8 = 34$, and so on. Apparently $b_n = f_{n+1} - 1$. We can prove this by mathematical induction. It is clearly true for small values of n. Assuming the inductive hypothesis, we have $b_n = b_{n-1} + b_{n-2} + 1 = (f_n - 1) + (f_{n-1} - 1) + 1 = f_n + f_{n-1} - 1 = f_{n+1} - 1$, as desired. Since b_n was just $\log a_n$, we know that $a_n = 2^{b_n} = 2^{f_{n+1}-1}$.

SECTION 9.2 Spanning Trees

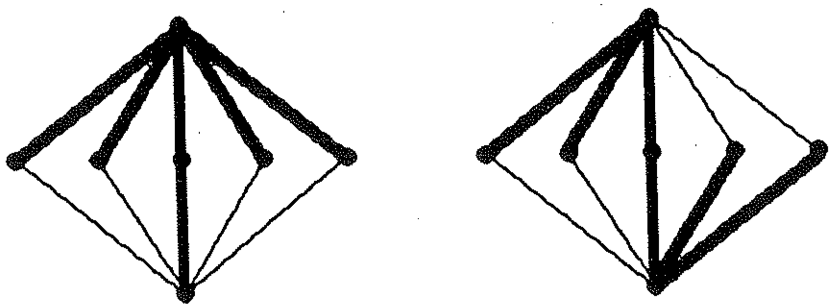
1. In each case we draw a picture of the graph and then darken enough edges to produce a spanning tree—to connect enough vertices so that there is only one component, without creating any cycles. We need to make sure that the trees we come up with are not isomorphic. These are certainly not the only possible answers.

   (a) The tree on the left is just a path; it has only vertices of degree 1 and 2. The tree on the right has a vertex of degree 3.





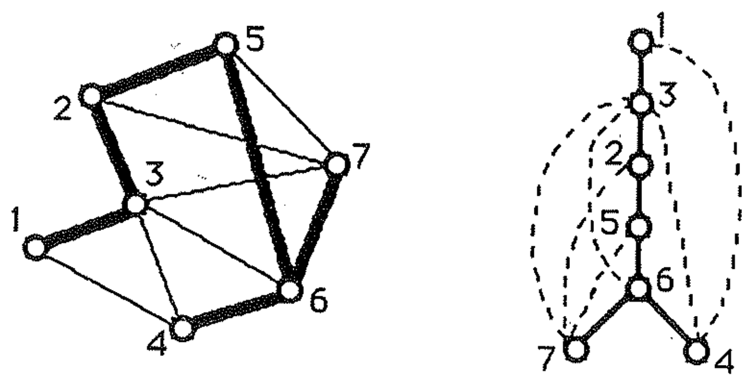
(b) The tree on the left has a vertex of degree 5, whereas the tree on the right does not.



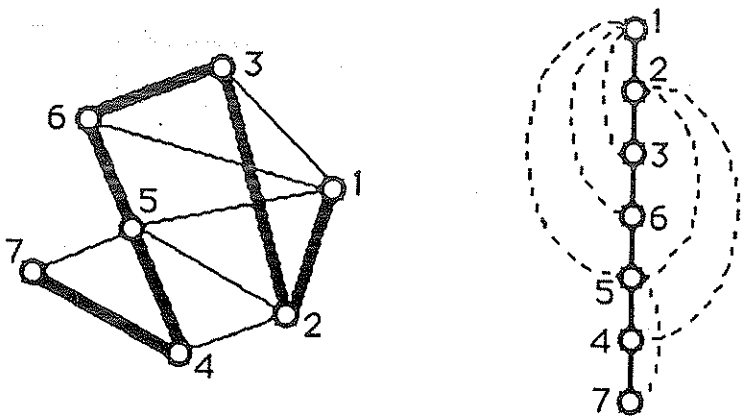
(c) The tree on the left is a star, with a vertex of degree 6; the tree on the right is a path with vertices having only degree 1 or 2.

3. In each case the figure shows the vertices numbered in the order specified. (There is no significance to the fact that the vertices are shown with hollow dots; they just contrast better with the heavy lines.)

(a) We start at vertex 1. The first vertex in numerical order adjacent to vertex 1 is vertex 3, so we put edge 13 into the tree and continue the construction from vertex 3. From vertex 3 we note that vertex 1 has already been visited, but that vertex 2 has not, so we put edge 32 into the tree and continue the search from vertex 2. We continue in this way, going to vertices 5, 6, and 4, adding edges 25, 56, and 64. Now as we process vertex 4 we see that there are no unvisited neighbors, so we are finished with vertex 4. Thus we return to the processing of vertex 6 and proceed to its last neighbor, vertex 7, adding edge 67 to the tree. Vertex 7 is adjacent to no unvisited vertices, so we backtrack to vertex 6, which has now been finished, then back to vertex 5, which is also finished, once we see that vertex 7 has already been visited, then back to 2, 3, and, finally, 1. Vertex 1 as well is adjacent to no more unvisited vertices, so the search is over. To redraw the tree in conventional form we put the root, vertex 1 (where we started the search), at the top. Note that we have drawn all the tree edges with solid lines, and the back edges (dashed lines) always connect vertices that stand in the ancestor–descendant relation.



(b) The process is basically the same as in part (a), again starting from the vertex labeled 1 and processing the unvisited neighbors of each vertex in numerical order, always completing the depth-first search of each new vertex that we visit before continuing with the vertex from which we reached the new vertex. Note that this time the tree turns out to be a path.



5. For the base case we note that if G has no edges, then G must be K_1, which is itself a tree. Assume that the statement is true for graphs with k edges. Let G be a graph with $k+1$ edges. If G is a tree, we are done. Otherwise, G must have a cycle. Remove an edge from the cycle. The resulting graph is still connected, but it has k edges. By the inductive hypothesis, it has a spanning tree, which perforce is also a spanning tree for G.

7. We need to choose a spanning tree in each component. In the component on the left we need to delete any one edge—four choices. For the component at the top we must delete any two of the edges, but not so as to disconnect the component. Of the $C(5,2) = 10$ choices for pairs of edges to delete, two of the pairs will leave isolated vertices; the other eight are acceptable. There is no choice involved in the other components. Therefore by the multiplication principle the answer is $4 \cdot 8 = 32$.