CHAPTER 10
Trees

SECTION 10.1 Introduction to Trees

2. a) This is a tree since it is connected and has no simple circuits.

b) This is a tree since it is connected and has no simple circuits.

c) This is not a tree, since it is not connected.

d) This is a tree since it is connected and has no simple circuits.

e) This is not a tree, since it has a simple circuit.

f) This is a tree since it is connected and has no simple circuits.

4. a) Vertex a is the root, since it is drawn at the top.

b) The internal vertices are the vertices with children, namely a, b, d, e, g, h, i, and o.

c) The leaves are the vertices without children, namely c, f, j, k, l, m, n, p, q, r, and s.

d) The children of j are the vertices adjacent to j and below j. There are no such vertices, so there are no children.

e) The parent of h is the vertex adjacent to h and above h, namely d.

f) Vertex o has only one sibling, namely p, which is the other child of o's parent, i.

g) The ancestors of m are all the vertices on the unique simple path from m back to the root, namely g, b, and a.

h) The descendants of b are all the vertices that have b as an ancestor, namely e, f, g, j, k, l, and m.

6. This is not a full m-ary tree for any m. It is an m-ary tree for all $m \geq 3$, since each vertex has at most 3 children, but since some vertices have 3 children, while others have 1 or 2, it is not full for any m.

8. We can easily determine the levels from the drawing. The root a is at level 0. The vertices in the row below a are at level 1, namely b, c, and d. The vertices below that, namely e through i (in alphabetical order), are at level 2. Similarly j through p are at level 3, and q, r, and s are at level 4.

10. We describe the answers, rather than actually drawing pictures.

a) The subtree rooted at a is the entire tree, since a is the root.

b) The subtree rooted at c consists of just the vertex c.

c) The subtree rooted at e consists of e, j, and k, and the edges ej and ek.

12. We find the answer by carefully enumerating these trees, i.e., drawing a full set of nonisomorphic trees. One way to organize this work so as to avoid leaving any trees out or counting the same tree (up to isomorphism) more than once is to list the trees by the length of their longest simple path (or longest simple path from the root in the case of rooted trees).

a) There are two trees with four vertices, namely $K_{1,3}$ and the simple path of length 3. See the first two trees below.

b) The longest path from the root can have length 1, 2 or 3. There is only one tree with longest path of length 1 (the other three vertices are at level 1), and only one with longest path of length 3. If the longest path has length 2, then the fourth vertex (after using three vertices to draw this path) can be "attached" to either the root or the vertex at level 1, giving us two nonisomorphic trees. Thus there are a total of four nonisomorphic rooted trees on 4 vertices, as shown below.

14. There are two things to prove. First suppose that T is a tree. By definition it is connected, so we need to show that the deletion of any of its edges produces a graph that is not connected. Let $\{x, y\}$ be an edge of T, and note that $x \neq y$. Now T with $\{x, y\}$ deleted has no path from x to y, since there was only one simple path from x to y in T, and the edge itself was it. (We use Theorem 1 here, as well as the fact that if there is a path from a vertex u to another vertex v, then there is a simple path from u to v by Theorem 1 in Section 9.4.) Therefore the graph with $\{x, y\}$ deleted is not connected.

Conversely, suppose that a simple connected graph T satisfies the condition that the removal of any edge will disconnect it. We must show that T is a tree. If not, then T has a simple circuit, say $x_1, x_2, \ldots, x_r, x_1$. If we delete edge $\{x_r, x_1\}$ from T, then the graph will remain connected, since wherever the deleted edge was used in forming paths between vertices we can instead use the rest of the circuit: $x_1, x_2, \ldots, x_r$ or its reverse, depending on which direction we need to go. This is a contradiction to the condition. Therefore our assumption was wrong, and T is a tree.

16. If both m and n are at least 2, then clearly there is a simple circuit of length 4 in $K_{m,n}$. On the other hand, $K_{m,1}$ is clearly a tree (as is $K_{1,n}$). Thus we conclude that $K_{m,n}$ is a tree if and only if $m = 1$ or $n = 1$.

18. By Theorem 4(ii), the answer is $mi + 1 = 5 \cdot 100 + 1 = 501$.

20. By Theorem 4(i), the answer is $[(m-1)n + 1]/m = (2 \cdot 100 + 1)/3 = 67$.

22. The model here is a full 5-ary tree. We are told that there are 10,000 internal vertices (these represent the people who send out the letter). By Theorem 4(ii) we see that $n = mi + 1 = 5 \cdot 10000 + 1 = 50{,}001$. Everyone but the root receives the letter, so we conclude that 50,000 people receive the letter. There are $50001 - 10000 = 40{,}001$ leaves in the tree, so that is the number of people who receive the letter but do not send it out.

24. Such a tree does exist. By Theorem 4(iii), we note that such a tree must have $i = 75/(m-1)$ internal vertices. This has to be a whole number, so $m - 1$ must divide 75. This is possible, for example, if $m = 6$, so let us try it. A complete 6-ary tree (see preamble to Exercise 27) of height 2 would have 36 leaves. We therefore need to add 40 leaves. This can be accomplished by changing 8 vertices at level 2 to internal vertices; each such change adds 5 leaves to the tree (6 new leaves at level 3, less the one leaf at level 5 that has been changed to an internal vertex). We will not show a picture of this tree, but just summarize its appearance. The root has 6 children, each of which has 6 children, giving 36 vertices at level 2. Of these, 28 are leaves, and each of the remaining 8 vertices at level 2 has 6 children, living at level 3, for a total of 48 leaves at level 3. The total number of leaves is therefore $28 + 48 = 76$, as desired.

26. By Theorem 4(iii), we note that such a tree must have $i = 80/(m-1)$ internal vertices. This has to be a whole number, so $m - 1$ must divide 80. By enumerating the divisors of 80, we see that m can equal 2, 3, 5, 6, 9, 11, 17, 21, 41, or 81. Some of these are incompatible with the height requirements, however.

a) Since the height is 4, we cannot have $m = 2$, since that will give us at most $1+2+4+8+16 = 31$ vertices. Any of the larger values of m shown above, up to 21, allows us to form a tree with 81 leaves and height 4. In each case we could get m^4 leaves if we made all vertices at levels smaller than 4 internal; and we can get as few as $4(m-1)+1$ leaves by putting only one internal vertex at each such level. We can get 81 leaves in the former case by taking $m = 3$; on the other hand, if $m > 21$, then we would be forced to have more than 81 leaves. Therefore the bounds on m are $3 \le m \le 21$ (with m also restricted to being in the list above).
b) If T must be balanced, then the smallest possible number of leaves is obtained when level 3 has only one internal vertex and $m^3 - 1$ leaves, giving a total of $m^3 - 1 + m$ leaves in T. Again, the maximum number of leaves will be m^4. With these restriction, we see that $m = 5$ is already too big, since this would require at least $5^3 - 1 + 5 = 129$ leaves. Therefore the only possibility is $m = 3$.

28. This tree has 1 vertex at level 0, m vertices at level 1, m^2 vertices at level 2, $\ldots$, m^h vertices at level h. Therefore it has

$$1 + m + m^2 + \cdots + m^h = \frac{m^{h+1} - 1}{m - 1}$$

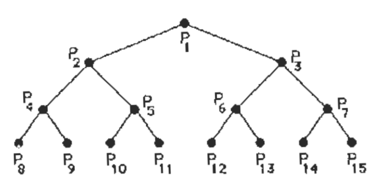
vertices in all. The vertices at level h are the only leaves, so it has m^h leaves.

30. (We assume $m \ge 2$.) First we delete all the vertices at level h; there is at least one such vertex, and they are all leaves. The result must be a complete m-ary tree of height $h-1$. By the result of Exercise 28, this tree has m^{h-1} leaves. In the original tree, then, there are more than this many leaves, since every internal vertex at level $h-1$ (which counts as a leaf in our reduced tree) spawns at least two leaves at level h.

32. The root of the tree represents the entire book. The vertices at level 1 represent the chapters—each chapter is a chapter of (read "child of") the book. The vertices at level 2 represent the sections (the parent of each such vertex is the chapter in which the section resides). Similarly the vertices at level 3 are the subsections.

34. a) The parent of a vertex is that vertex's boss.
b) The child of a vertex is an immediate subordinate of that vertex (one he or she directly supervises).
c) The sibling of a vertex is a coworker with the same boss.
d) The ancestors of a vertex are that vertex's boss, his/her boss's boss, etc.
e) The descendants of a vertex are all the people that that vertex ultimately supervises (directly or indirectly).
f) The level of a vertex is the number of levels away from the top of the organization that vertex is.
g) The height of the tree is the depth of the structure.

36. a) We simply add one more row to the tree in Figure 12, obtaining the following tree.



b) During the first step we use the bottom row of the network to add $x_1 + x_2$, $x_3 + x_4$, $x_5 + x_6$, $\ldots$, $x_{15} + x_{16}$. During the second step we use the next row up to add the results of the computations from the first step, namely $(x_1 + x_2) + (x_3 + x_4)$, $(x_5 + x_6) + (x_7 + x_8)$, $\ldots$, $(x_{13} + x_{14}) + (x_{15} + x_{16})$. The third step uses the sums obtained in the second, and the two processors in the second row of the tree perform $(x_1 + x_2 + x_3 + x_4) + (x_5 + x_6 + x_7 + x_8)$ and $(x_9 + x_{10} + x_{11} + x_{12}) + (x_{13} + x_{14} + x_{15} + x_{16})$. Finally, during the fourth step the root processor adds these two quantities to obtain the desired sum.

38. For $n = 3$, there is only one tree to consider, the one that is a simple path of length 2. There are 3 choices for the label to put in the middle of the path, and once that choice is made, the labeled tree is determined up to isomorphism. Therefore there are 3 labeled trees with 3 vertices.

For $n = 4$, there are two structures the tree might have. If it is a simple path with length 3, then there are 12 different labelings; this follows from the fact that there are $P(4,4) = 4! = 24$ permutations of the integers from 1 to 4, but a permutation and its reverse lead to the same labeled tree. If the tree structure is $K_{1,3}$, then the only choice is which label to put on the vertex that is adjacent to the other three, so there are 4 such trees. Thus in all there are 16 labeled trees with 4 vertices.

In fact it is a theorem that the number of labeled trees with n vertices is n^{n-2} for all $n \geq 2$.

40. The eccentricity of vertex e is 3, and it is the only vertex with eccentricity this small. Therefore e is the only center.

42. Since the height of a tree is the maximum distance from the root to another vertex, this is clear from the definition of center.

44. We choose a root and color it red. Then we color all the vertices at odd levels blue and all the vertices at even levels red.

46. The number of vertices in the tree T_n satisfies the recurrence relation $v_n = v_{n-1} + v_{n-2} + 1$ (the "$+1$" is for the root), with $v_1 = v_2 = 1$. Thus the sequence begins $1, 1, 3, 5, 9, 15, 25, \ldots$. It is easy to prove by induction that $v_n = 2f_n - 1$, where f_n is the n^{th} Fibonacci number. The number of leaves satisfies the recurrence relation $l_n = l_{n-1} + l_{n-2}$, with $l_1 = l_2 = 1$, so $l_n = f_n$. Since $i_n = v_n - l_n$, we have $i_n = f_n - 1$. Finally, it is clear that the height of the tree T_n is one more than the height of the tree T_{n-1} for $n \geq 3$, with the height of T_2 being 0. Therefore the height of T_n is $n - 2$ for all $n \geq 2$ (and of course the height of T_1 is 0).

48. Let T be a tree with n vertices, having height h. If there are any internal vertices in T at levels less than $h - 1$ that do not have two children, take a leaf at level h and move it to be such a missing child. This only lowers the average depth of a leaf in this tree, and since we are trying to prove a lower bound on the average depth, it suffices to prove the bound for the resulting tree. Repeat this process until there are no more internal vertices of this type. As a result, all the leaves are now at levels $h - 1$ and h. Now delete all vertices at level h. This changes the number of vertices by at most (one more than) a factor of two and so has no effect on a big-Omega estimate (it changes $\log n$ by at most 1). Now the tree is complete, and by Exercise 28 it has 2^{h-1} leaves, all at depth $h - 1$, where now $n = 2^h - 1$. The desired estimate follows.