TREES

A *tree* is a connected graph without any circuits. The graph in Fig. 3-1, for instance, is a tree. Trees with one, two, three and four vertices are shown in Fig. 3-2. As a graph must have at least one vertex and therefore so must a tree. Some authors allow the null tree, a tree without any vertices. We have excluded such an entity from being a tree. Similarly as we are considering only finite graphs, our trees are also finite.

It follows immediately from the definition that a tree has to be a simple graph, that is, having neither a self-loop nor parallel edges (because they both form circuits).

Trees appear in numerous instances. The genealogy of a family is often

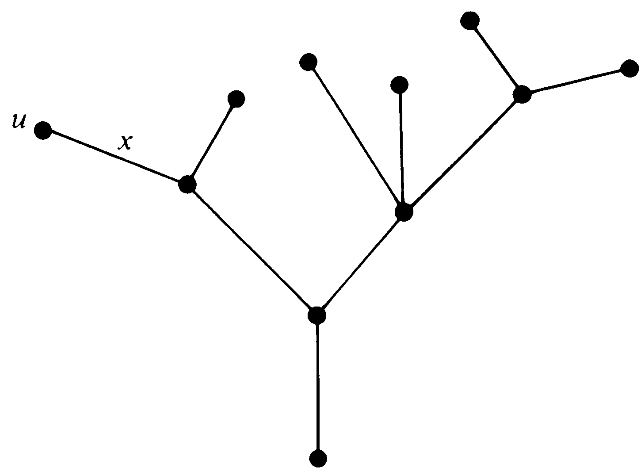

Fig. 3-1 Tree.

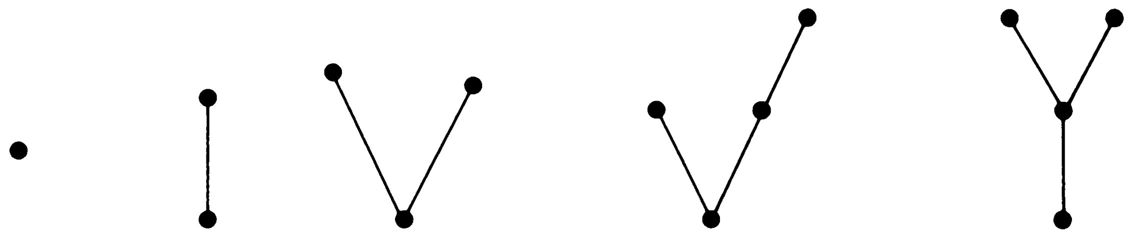

Fig. 3-2 Trees with one, two, three, and four vertices.

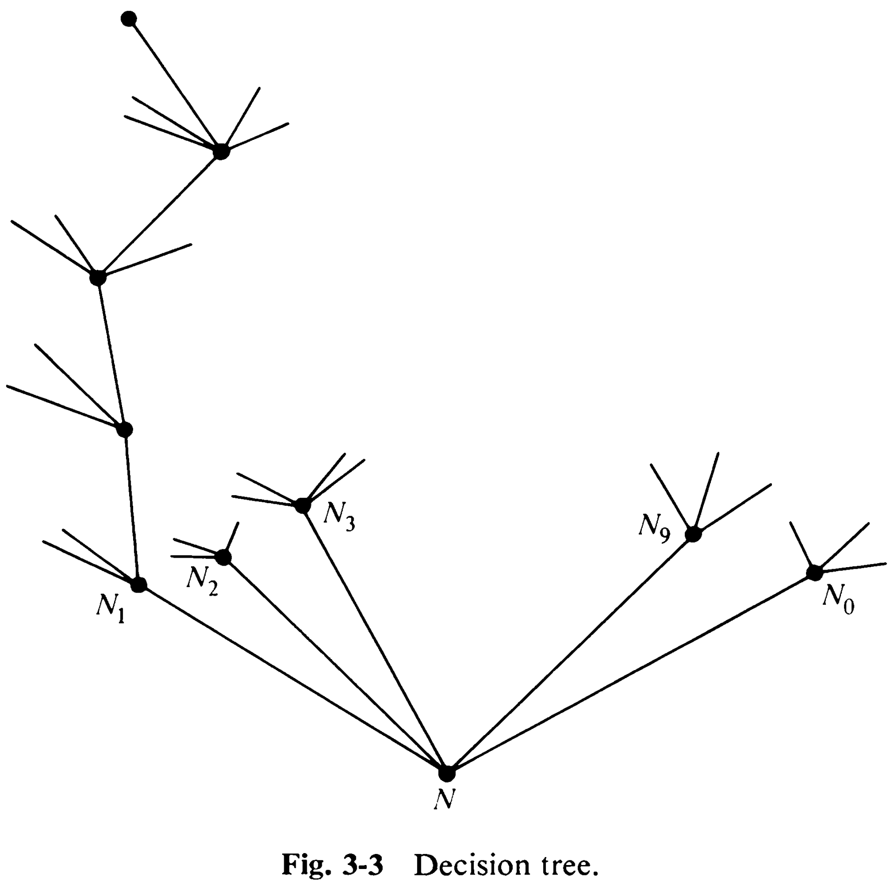

Fig. 3-3 Decision tree.

represented by means of a tree (in fact the term *tree* comes from *family tree*). A river with its tributaries and subtributaries can be represented by a tree. The sorting of mail according to zip code and the sorting of punched cards are done according to a tree (called *decision tree* or *sorting tree*).

Figure 3-3 might represent the flow of mail. All the mail arrives at some local office, vertex N. The most significant digit in the zip code is read at N, and the mail is divided into 10 piles $N_1, N_2, \ldots, N_9$, and N_0, depending on the most significant digit. Each pile is further divided into 10 piles according to the second most significant digit, and so on, till the mail is subdivided into 10^5 possible piles, each representing a unique five-digit zip code.

In many sorting problems we have only two alternatives (instead of 10 as in the preceding example) at each intermediate vertex, representing a dichotomy, such as large or small, good or bad, 0 or 1. Such a decision tree with two choices at each vertex occurs frequently in computer programming and switching theory. We shall deal with such trees and their applications in Section 3-5. Let us first obtain a few simple but important theorems on the general properties of trees.

3-2. SOME PROPERTIES OF TREES

THEOREM 3-1

There is one and only one path between every pair of vertices in a tree, T.

Proof: Since T is a connected graph, there must exist at least one path between every pair of vertices in T. Now suppose that between two vertices a and b of T there are two distinct paths. The union of these two paths will contain a circuit and T cannot be a tree. ■

Conversely:

THEOREM 3-2

If in a graph G there is one and only one path between every pair of vertices, G is a tree.

Proof: Existence of a path between every pair of vertices assures that G is connected. A circuit in a graph (with two or more vertices) implies that there is at least one pair of vertices a, b such that there are two distinct paths between a and b. Since G has one and only one path between every pair of vertices, G can have no circuit. Therefore, G is a tree. ■

THEOREM 3-3

A tree with n vertices has $n - 1$ edges.

Proof: The theorem will be proved by induction on the number of vertices.

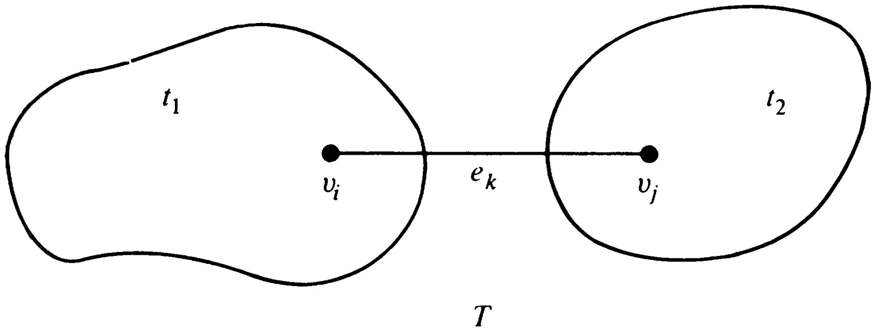

Fig. 3-4 Tree T with n vertices.

It is easy to see that the theorem is true for $n = 1, 2,$ and 3 (see Fig. 3-2). Assume that the theorem holds for all trees with fewer than n vertices.

Let us now consider a tree T with n vertices. In T let e_k be an edge with end vertices v_i and v_j. According to Theorem 3-1, there is no other path between v_i and v_j except e_k. Therefore, deletion of e_k from T will disconnect the graph, as shown in Fig. 3-4. Furthermore, $T - e_k$ consists of exactly two components, and since there were no circuits in T to begin with, each of these components is a tree. Both these trees, t_1 and t_2, have fewer than n vertices each, and therefore, by the induction hypothesis, each contains one less edge than the number of vertices in it. Thus $T - e_k$ consists of $n - 2$ edges (and n vertices). Hence T has exactly $n - 1$ edges. ■

THEOREM 3-4

Any connected graph with n vertices and $n - 1$ edges is a tree.

Proof: The proof of the theorem is left to the reader as an exercise (Problem 3-5).

You may have noticed another important feature of a tree: its vertices are connected together with the minimum number of edges. A connected graph is said to be *minimally connected* if removal of any one edge from it disconnects the graph. A minimally connected graph cannot have a circuit; otherwise, we could remove one of the edges in the circuit and still leave the graph connected. Thus a minimally connected graph is a tree. Conversely, if a connected graph G is not minimally connected, there must exist an edge e_i in G such that $G - e_i$ is connected. Therefore, e_i is in some circuit, which implies that G is not a tree. Hence the following theorem:

THEOREM 3-5

A graph is a tree if and only if it is minimally connected.

The significance of Theorem 3-5 is obvious. Intuitively, one can see that to interconnect n distinct points, the minimum number of line segments needed is $n - 1$. It requires no background in electrical engineering to realize

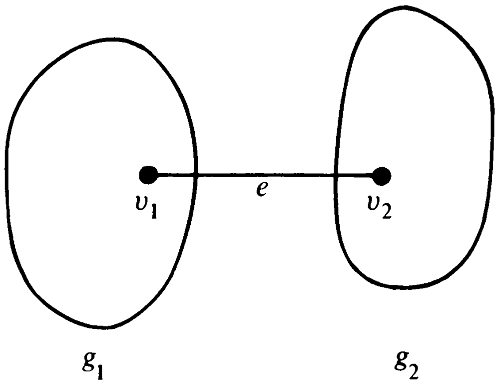

Fig. 3-5 Edge e added to $G = g_1 \cup g_2$.

that to short (electrically) n pins together, one needs at least $n - 1$ pieces of wire. The resulting structure, according to Theorem 3-5, is a tree.

We showed that a connected graph with n vertices and without any circuits has $n - 1$ edges. We can also show that a graph with n vertices which has no circuit and has $n - 1$ edges is always connected (i.e., it is a tree), in the following theorem.

THEOREM 3-6

A graph G with n vertices, $n - 1$ edges, and no circuits is connected.

Proof: Suppose there exists a circuitless graph G with n vertices and $n - 1$ edges which is disconnected. In that case G will consist of two or more circuitless components. Without loss of generality, let G consist of two components, g_1 and g_2. Add an edge e between a vertex v_1 in g_1 and v_2 in g_2 (Fig. 3-5). Since there was no path between v_1 and v_2 in G, adding e did not create a circuit. Thus $G \cup e$ is a circuitless, connected graph (i.e., a tree) of n vertices and n edges, which is not possible, because of Theorem 3-3. ■

The results of the preceding six theorems can be summarized by saying that the following are five different but equivalent definitions of a tree. That is, a graph G with n vertices is called a tree if

1. G is *connected* and is *circuitless*, or
2. G is *connected* and has $n - 1$ *edges*, or
3. G is *circuitless* and has $n - 1$ *edges*, or
4. There is *exactly one path* between every pair of vertices in G, or
5. G is a *minimally connected* graph.

3-3. PENDANT VERTICES IN A TREE

You must have observed that each of the trees shown in the figures has several pendant vertices (a pendant vertex was defined as a vertex of degree

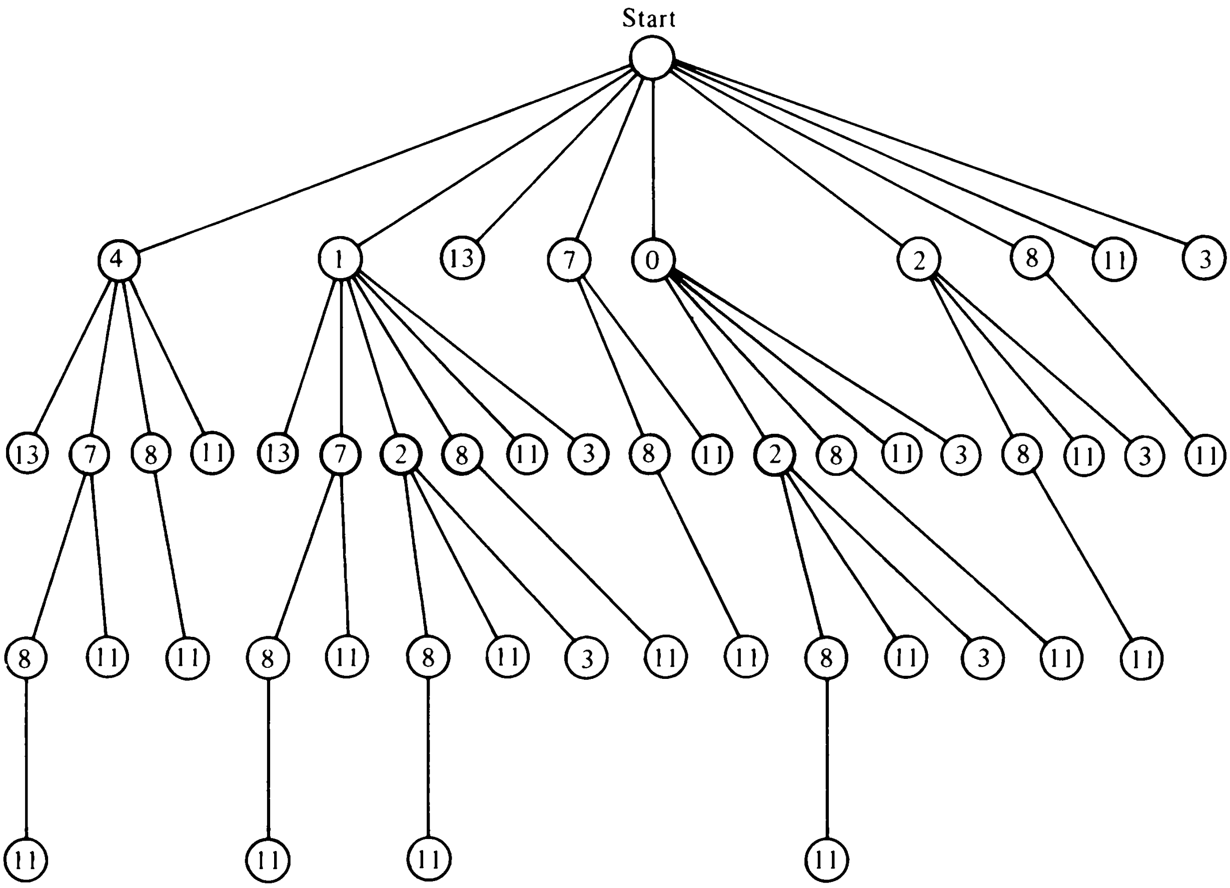

Fig. 3-6 Tree of the monotonically increasing sequences in 4, 1, 13, 7, 0, 2, 8, 11, 3.

one). The reason is that in a tree of n vertices we have $n - 1$ edges, and hence $2(n - 1)$ degrees to be divided among n vertices. Since no vertex can be of zero degree, we must have at least two vertices of degree one in a tree. This of course makes sense only if $n \geq 2$. More formally:

THEOREM 3-7

In any tree (with two or more vertices), there are at least two pendant vertices.

An Application: The following problem is used in teaching computer programming. Given a sequence of integers, no two of which are the same, find the largest monotonically increasing subsequence in it. Suppose that the sequence given to us is 4, 1, 13, 7, 0, 2, 8, 11, 3; it can be represented by a tree in which the vertices (except the start vertex) represent individual numbers in the sequence, and the path from the start vertex to a particular vertex v describes the monotonically increasing subsequence terminating in v. As shown in Fig. 3-6, this sequence contains four longest monotonically increasing subsequences, that is, (4, 7, 8, 11), (1, 7, 8, 11), (1, 2, 8, 11), and (0, 2, 8, 11). Each is of length four. Such a tree used in representing data is referred to as a data tree by computer programmers.

3-4. DISTANCE AND CENTERS IN A TREE

The tree in Fig. 3-7 has four vertices. Intuitively, it seems that vertex b is located more "centrally" than any of the other three vertices. We shall ex-

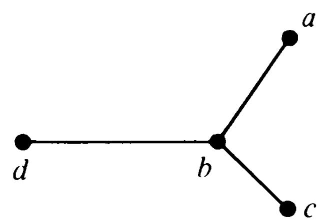

Fig. 3-7 Tree.

plore this idea further and see if in a tree there exists a "center" (or centers). Inherent in the concept of a center is the idea of "distance," so we must define distance before we can talk of a center.

In a connected graph G, the *distance* $d(v_i, v_j)$ between two of its vertices v_i and v_j is the length of the shortest path (i.e., the number of edges in the shortest path) between them.

The definition of distance between any two vertices is valid for any connected graph (not necessarily a tree). In a graph that is not a tree, there are generally several paths between a pair of vertices. We have to enumerate all these paths and find the length of the shortest one. (There may be several shortest paths.)

For instance, some of the paths between vertices v_1 and v_2 in Fig. 3-8 are (a, e), (a, c, f), (b, c, e), (b, f), (b, g, h), and (b, g, i, k). There are two shortest paths, (a, e) and (b, f), each of length two. Hence $d(v_1, v_2) = 2$.

In a tree, since there is exactly one path between any two vertices (Theorem 3-1), the determination of distance is much easier. For instance, in the tree of Fig. 3-7, $d(a, b) = 1$, $d(a, c) = 2$, $d(c, b) = 1$, and so on.

A Metric: Before we can legitimately call a function $f(x, y)$ of two variables a "distance" between them, this function must satisfy certain requirements. These are

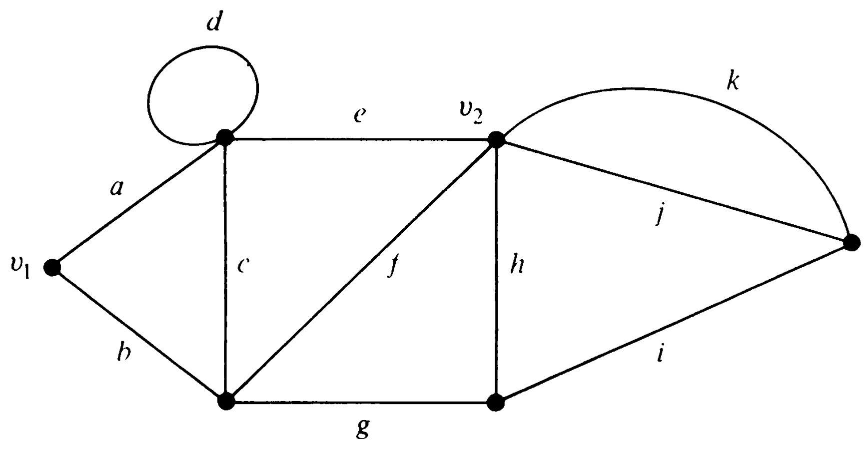

Fig. 3-8 Distance between v_1 and v_2 is two.

1. Nonnegativity: $f(x, y) \geq 0$, and $f(x, y) = 0$ if and only if $x = y$.
2. Symmetry: $f(x, y) = f(y, x)$.
3. Triangle inequality: $f(x, y) \leq f(x, z) + f(z, y)$ for any z.

A function that satisfies these three conditions is called a *metric*. That the distance $d(v_i, v_j)$ between two vertices of a connected graph satisfies conditions 1 and 2 is immediately evident. Since $d(v_i, v_j)$ is the length of the shortest path between vertices v_i and v_j, this path cannot be longer than another path between v_i and v_j, which goes through a specified vertex v_k. Hence $d(v_i, v_j) \leq d(v_i, v_k) + d(v_k, v_j)$. Therefore,

THEOREM 3-8

The distance between vertices of a connected graph is a metric.

Coming back to our original topic of relative location of different vertices in a tree, let us define another term called *eccentricity* (also referred to as *associated number* or *separation*) of a vertex in a graph.

The eccentricity $E(v)$ of a vertex v in a graph G is the distance from v to the vertex farthest from v in G; that is,

$$E(v) = \max_{v_i \in G} d(v, v_i).$$

A vertex with minimum eccentricity in graph G is called a *center* of G. The eccentricities of the four vertices in Fig. 3-7 are $E(a) = 2$, $E(b) = 1$, $E(c) = 2$, and $E(d) = 2$. Hence vertex b is the center of that tree. On the other hand, consider the tree in Fig. 3-9. The eccentricity of each of its six vertices is shown next to the vertex. This tree has two vertices having the same minimum eccentricity. Hence this tree has two centers. Some authors refer to such centers as *bicenters;* we shall call them just centers, because there will be no occasion for confusion.

The reader can easily verify that a graph, in general, has many centers. For example, in a graph that consists of just a circuit (a polygon), every vertex is a center. In the case of a tree, however, König [1-7] proved the following theorem:

THEOREM 3-9

Every tree has either one or two centers.

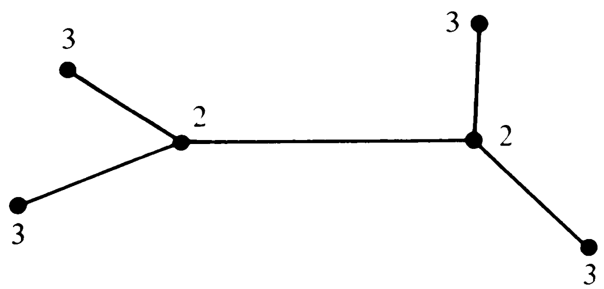

Fig 3-9 Eccentricities of the vertices of a tree.

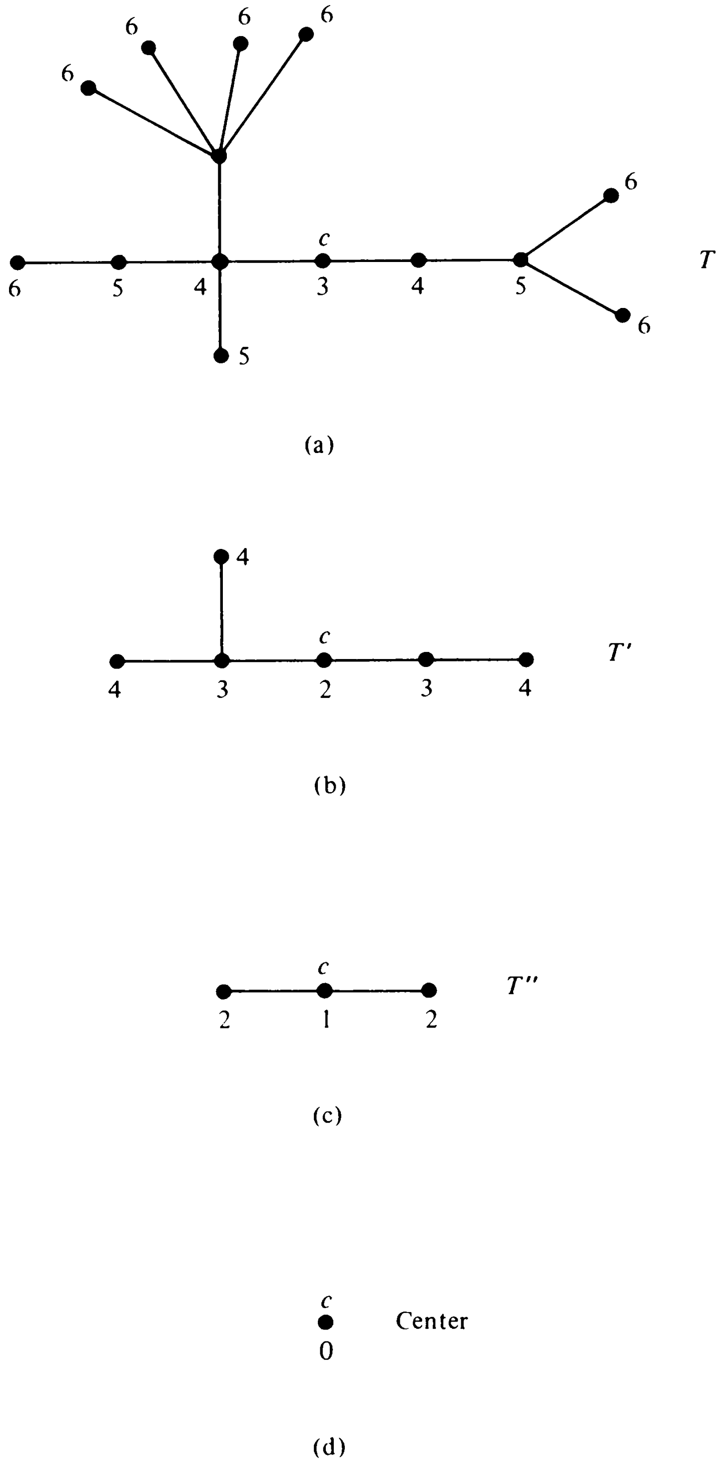

Fig. 3-10 Finding a center of a tree.

Proof: The maximum distance, max $d(v, v_i)$, from a given vertex v to any other vertex v_i occurs only when v_i is a pendant vertex. With this observation, let us start with a tree T having more than two vertices. Tree T must have two or more pendant vertices (Theorem 3-7). Delete all the pendant vertices from T. The resulting graph T' is still a tree. What about the eccentricities of the vertices in T'? A little deliberation will reveal that removal of all pendant vertices from T uniformly reduced the eccentricities of the remaining vertices (i.e., vertices in T') by one. Therefore, all vertices that T had as centers will still remain centers in T'. From T' we can again remove all pendant vertices and get another tree T''. We continue this process (which is illustrated in Fig. 3-10) until there is left either a vertex (which is the center of T) or an edge (whose end vertices are the two centers of T). Thus the theorem. ■