Discrete Mathematics

CSE 106: Department of Mathematics

Dr. Mohammad Salah Uddin

Associate Professor

East West University, Bangladesh

Introduction to Trees

- A connected graph that contains no simple circuits is called a tree.
- Trees have been employed to solve problems in a wide variety of disciplines, as the examples in this chapter will show.
- Trees are particularly useful in computer science, where they are employed in a wide range of algorithms. For instance, trees are used to construct efficient algorithms for locating items in a list.
- we will focus on a particular type of graph called a tree, so named because such graphs resemble trees. For example, family trees are graphs that represent genealogical charts. Family trees use vertices to represent the members of a family and edges to represent parent. child relationships.

DEFINITION 1:

A **tree** is a connected undirected graph with no simple circuits.

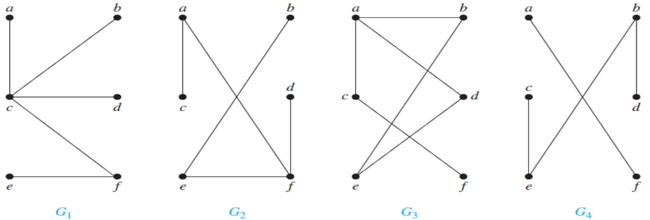

Figure 1: Examples of Trees and Graphs That Are Not Trees.

Example 1:

Which of the graphs shown in Figure 1 are trees?

Solution: G_1 and G_2 are trees, because both are connected graphs with no simple circuits. G_3 is not a tree because e, b, a, d, e is a simple circuit in this graph. Finally, G_4 is not a tree because it is not connected.

Introduction to Trees

Any connected graph that contains no simple circuits is a tree. What about graphs containing no simple circuits that are not necessarily connected? These graphs are called **forests** and have the property that each of their connected components is a tree. Figure 2 displays a forest.

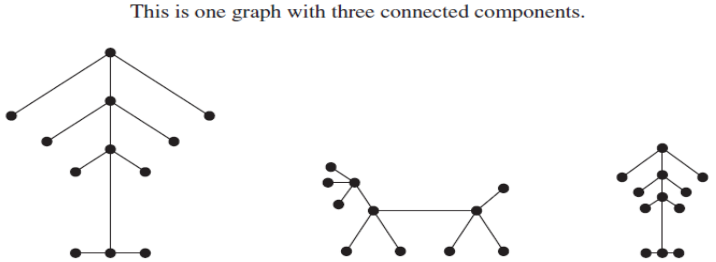

Figure 2: Example of a Forest.

Trees are often defined as undirected graphs with the property that there is a unique simple path between every pair of vertices. Theorem 1 shows that this alternative definition is equivalent to our definition.

THEOREM 1:

An undirected graph is a tree if and only if there is a unique simple path between any two of its vertices.

Rooted (Directed) Trees: In many applications of trees, a particular vertex of a tree is designated as the **root**. Once we specify a root, we can assign a direction to each edge as follows. Because there is a unique path from the root to each vertex of the graph (by Theorem 1), we direct each edge away from the root. Thus, a tree together with its root produces a directed graph called a **rooted tree**.

DEFINITION 2:

A **rooted tree** is a tree in which one vertex has been designated as the **root** and every edge is directed away from the root.

We can change an unrooted tree into a rooted tree by choosing any vertex as the root. Different choices of the root produce different rooted trees. For instance, Figure 3 displays the rooted trees formed by designating a to be the root and c to be the root, respectively, in the tree T .

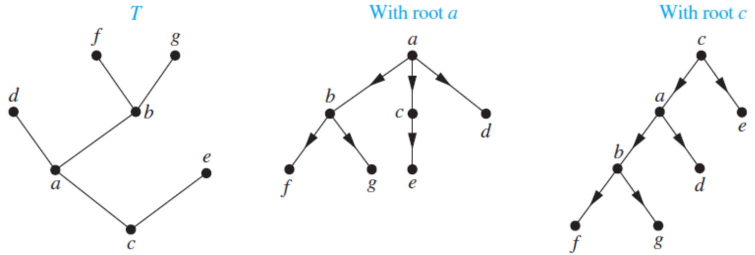

Figure 3: A Tree and Rooted Trees Formed by Designating Two Different Roots.

Introduction to Trees

The terminology for trees has botanical and genealogical origins. Suppose that T is a rooted tree. If v is a vertex in T other than the root

- the **parent** of v is the unique vertex u such that there is a directed edge from u to v.
- When u is the parent of v, v is called a **child** of u.
- Vertices with the same parent are called **siblings**.
- The **ancestors** of a vertex other than the root are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root.
- The **descendants** of a vertex v are those vertices that have v as an ancestor.
- A vertex of a rooted tree is called a **leaf** if it has no children.
- Vertices that have children are called **internal vertices**.
- If a is a vertex in a tree, the **subtree** with a as its root is the subgraph of the tree consisting of a and its descendants and all edges incident to these descendants.

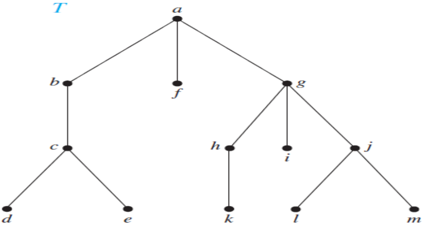

Figure 4: A Rooted Tree T.

Example 2:

In the rooted tree T (with root a) shown in Figure 4, find the parent of c, the children of g, the siblings of h, all ancestors of e, all descendants of b, all internal vertices, and all leaves. What is the subtree rooted at g?

Solution: The parent of c is b. The children of g are h, i, and j. The siblings of h are i and j. The ancestors of e are c, b, and a. The descendants of b are c, d, and e. The internal vertices are a, b, c, g, h, and j. The leaves are d, e, f, i, k, l, and m. The subtree rooted at g is shown in Figure 5.

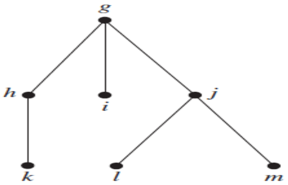

Figure 5: The Subtree Rooted at g.

DEFINITION:

- **Level (of v)** is the length of the unique path from the root to v.
- **Height** is maximum of vertices levels.

DEFINITION 3:

A rooted tree is called an **m-ary tree** if every internal vertex has no more than m children. The tree is called a **full m-ary tree** if every internal vertex has exactly m children. An m-ary tree with $m = 2$ is called a **binary tree**.

Example 3:

Are the rooted trees in Figure 6 full m-ary trees for some positive integer m?

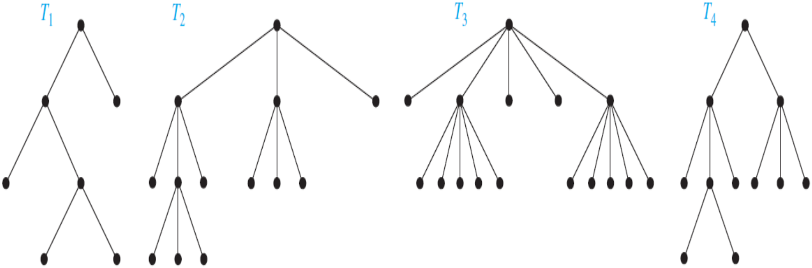

Figure 6: Rooted trees.

Solution: T_1 is a full binary tree because each of its internal vertices has two children. T_2 is a full 3-ary tree because each of its internal vertices has three children. In T_3 each internal vertex has five children, so T_3 is a full 5-ary tree. T_4 is not a full m-ary tree for any m because some of its internal vertices have two children and others have three children.

ORDERED ROOTED TREES

- An **ordered rooted tree** is a rooted tree where the children of each internal vertex are ordered.
- Ordered rooted trees are drawn so that the children of each internal vertex are shown in order from left to right.
- In an ordered binary tree (usually called just a binary tree), if an internal vertex has two children, the first child is called the **left child** and the second child is called the **right child**.
- The tree rooted at the left child of a vertex is called the **left subtree** of this vertex, and the tree rooted at the right child of a vertex is called the **right subtree** of the vertex.
- For m-ary trees with $m > 2$,we can use terms like **leftmost**, **rightmost**, etc.

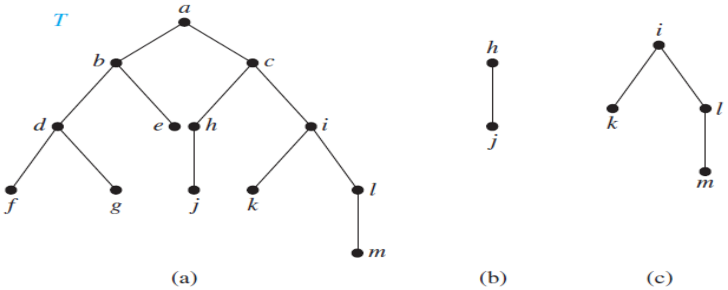

Figure 7: Rooted trees.

Example 4:

What are the left and right children of d in the binary tree T shown in Figure 7(a) (where the order is that implied by the drawing)? What are the left and right subtrees of c?

Solution: The left child of d is f and the right child is g. We show the left and right subtrees of c in Figures 7(b) and 7(c), respectively.

Trees as Models

Trees are used as models in such diverse areas as computer science, chemistry, geology, botany, and psychology.We will describe a variety of such models based on trees.

Example 5: Saturated Hydrocarbons and Trees

Graphs can be used to represent molecules, where atoms are represented by vertices and bonds between them by edges.

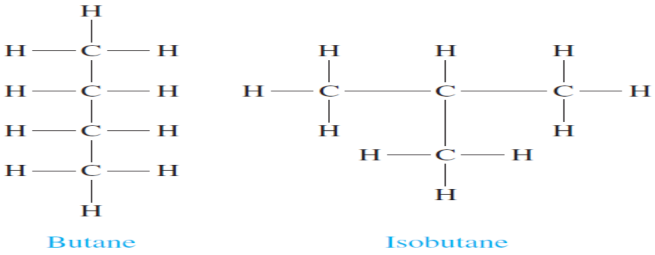

Figure 8: The Two Isomers of Butane.

Example 6: Representing Organizations

The structure of a large organization can be modeled using a rooted tree. Each vertex in this tree represents a position in the organization. An edge from one vertex to another indicates that the person represented by the initial vertex is the (direct) boss of the person represented by the terminal vertex.

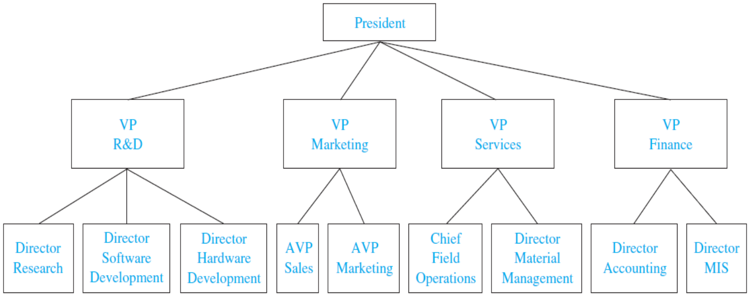

Figure 9: An Organizational Tree for a Computer Company.

Example 7: Computer File Systems

Files in computer memory can be organized into directories. A directory can contain both files and subdirectories. The root directory contains the entire file system. Thus, a file system may be represented by a rooted tree, where the root represents the root directory, internal vertices represent subdirectories, and leaves represent ordinary files or empty directories

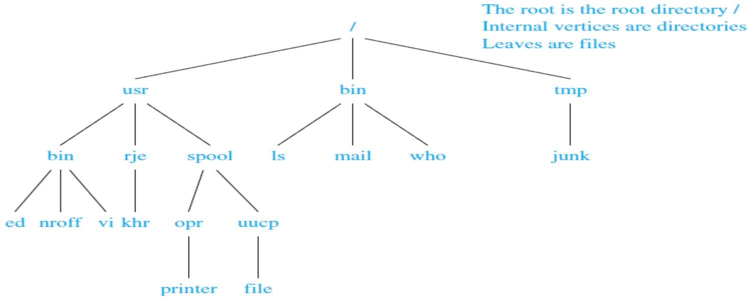

Figure 10: A Computer File System.

Properties of Trees

We will often need results relating the numbers of edges and vertices of various types in trees.

THEOREM 2:

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

THEOREM 3:

A full m-ary tree with i internal vertices and l leaves contains $n = mi + 1$ ($n = i + l$) vertices.

THEOREM 4:

A full m-ary tree with

1. n vertices has $i = (n-1)/m$ internal vertices and $l = [(m-1)n+1]/m$ leaves,
2. i internal vertices has $n = mi+1$ vertices and $l = (m-1)i+1$ leaves,
3. l leaves has $n = (ml-1)/(m-1)$ vertices and $i = (l-1)/(m-1)$ internal vertices.

Example 8 (9 in book):

Suppose that someone starts a chain letter. Each person who receives the letter is asked to send it on to four other people. Some people do this, but others do not send any letters. How many people have seen the letter, including the first person, if no one receives more than one letter and if the chain letter ends after there have been 100 people who read it but did not send it out? How many people sent out the letter?

Solution: The chain letter can be represented using a 4-ary tree. The internal vertices correspond to people who sent out the letter, and the leaves correspond to people who did not send it out. Because 100 people did not send out the letter, the number of leaves in this rooted tree is $l = 100$. Hence, part (3) of Theorem 4 shows that the number of people who have seen the letter is $n = (4 \times 100 - 1)/(4 - 1) = 133$. Also, the number of internal vertices is $133 - 100 = 33$, so 33 people sent out the letter.