ITT202: Logic and Graph Theory

Module 4: Trees

Instructor: Dr. Iqra Altaf Gillani

Introduction to Trees

- A connected graph that contains no simple circuits is called a tree.
- Trees were used as long ago as 1857, when the English mathematician Arthur Cayley used them to count certain types of chemical compounds.
- Trees are particularly useful in computer science, where they are employed in a wide range of algorithms.

Introduction to Trees

- A connected graph that contains no simple circuits is called a tree.
- 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.
- They can be used in algorithms, such as Huffman coding, that construct efficient codes saving costs in data transmission and storage.
- Trees can be used to study games such as checkers and chess and can help determine winning strategies for playing these games.
- Trees can be used to model procedures carried out using a sequence of decisions. Constructing these models can help determine the computational complexity of algorithms based on a sequence of decisions, such as sorting algorithms.

Example: Tree

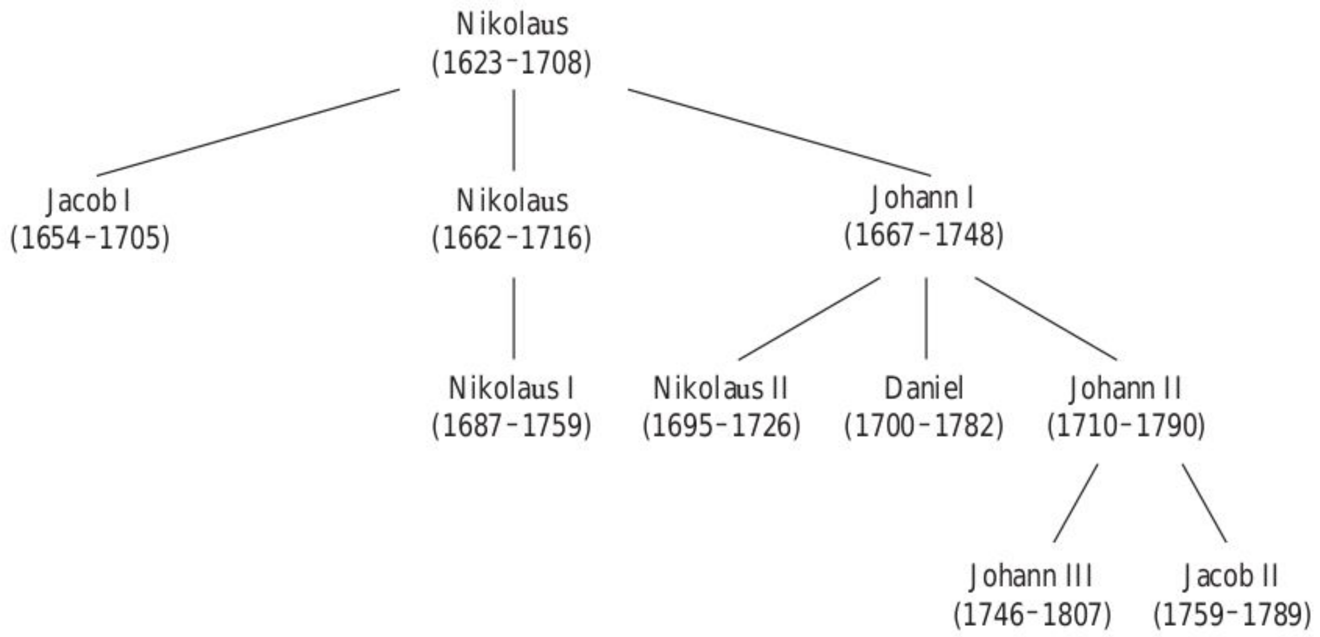


FIGURE 1 **The Bernoulli Family of Mathematicians.**

Tree

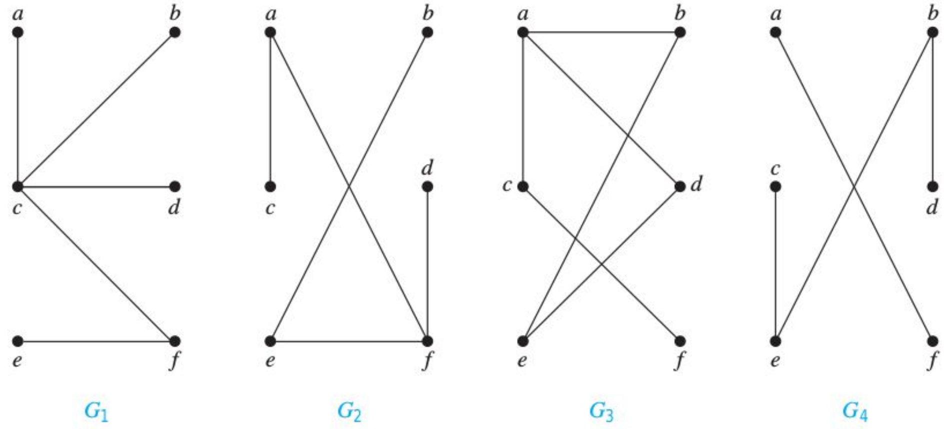
- A tree is a connected undirected graph with no simple circuits.
- Because a tree cannot have a simple circuit, a tree cannot contain multiple edges or loops. Therefore any tree must be a simple graph.

Tree

- A tree is a connected undirected graph with no simple circuits.
- Because a tree cannot have a simple circuit, a tree cannot contain multiple edges or loops. Therefore any tree must be a simple graph.
- *Theorem 1: An undirected graph is a tree if and only if there is a unique simple path between any two of its vertices.*

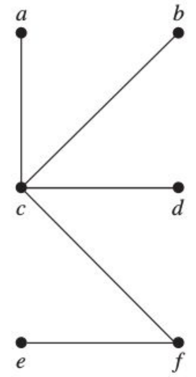
Tree

- A tree is a connected undirected graph with no simple circuits.
- Because a tree cannot have a simple circuit, a tree cannot contain multiple edges or loops. Therefore any tree must be a simple graph.
- Examples:

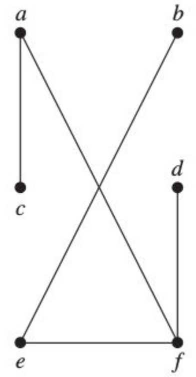


Tree

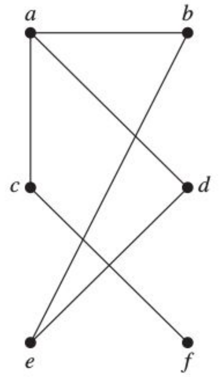
- A tree is a connected undirected graph with no simple circuits.
- Because a tree cannot have a simple circuit, a tree cannot contain multiple edges or loops. Therefore any tree must be a simple graph.
- Examples:



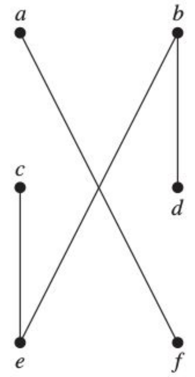
G_1



G_2



G_3



G_4

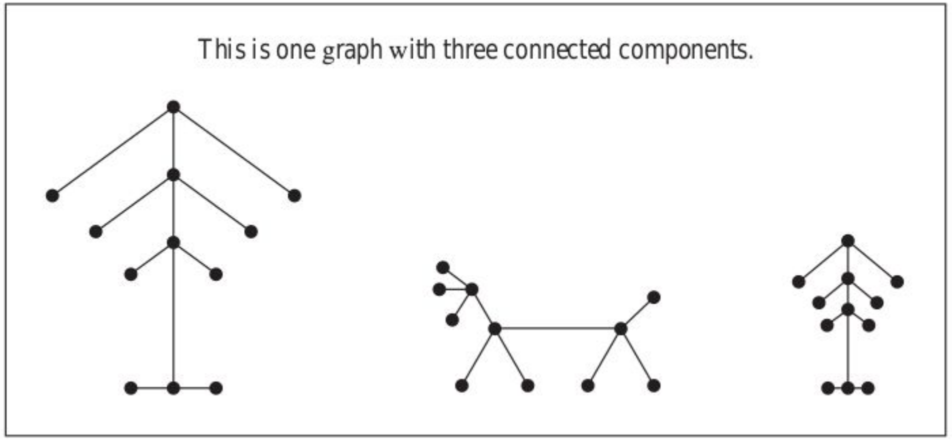
Solution: G 1 and G 2 are trees, because both are connected graphs with no simple circuits. G 3 is not a tree because e, b, a, d, e is a simple circuit in this graph. Finally, G 4 is not a tree because it is not connected.

Forests

- Any connected graph that contains no simple circuits is a tree. What about graphs containing no simple circuits that are not necessarily connected?

Forests

- 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.
- Example:

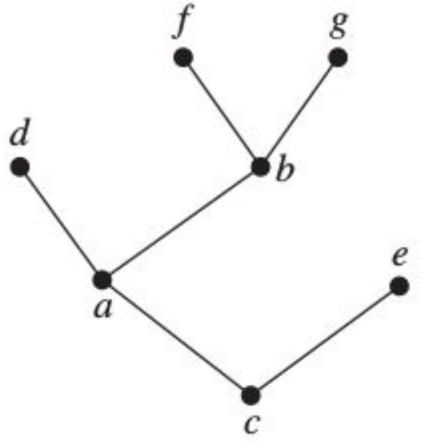


Rooted Trees

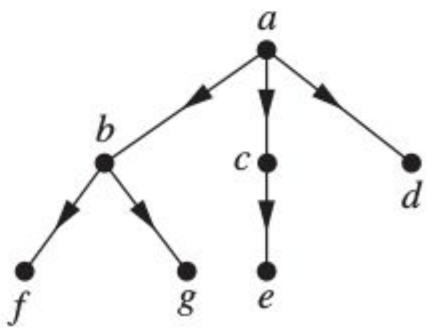
- In many applications of trees, a particular vertex of a tree is designated as the root. Once we specify a root, we can assign a direction to each edge as follows. Because there is a unique path from the root to each vertex of the graph (by Theorem 1), we direct each edge away from the root. Thus, a tree together with its root produces a directed graph called a rooted tree.
- *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.*
- Rooted trees can also be defined recursively.
- We can also change an unrooted tree into a rooted tree by choosing any vertex as the root. Note that different choices of the root produce different rooted trees.

Examples: Rooted Trees

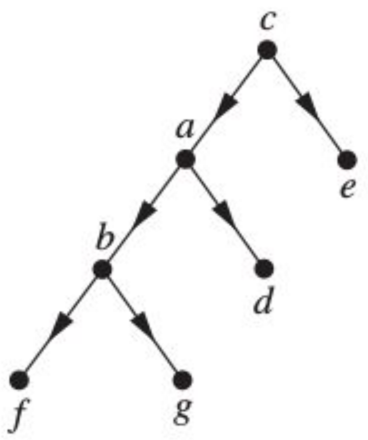
T



With root *a*



With root *c*



Terminologies

- 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 (the reader should show that such a vertex is unique).
- 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 (that is, its parent, its parent’s parent, and so on, until the root is reached).
- The descendants of a vertex v are those vertices that have v as an ancestor.

Terminologies

- 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. The root is an internal vertex unless it is the only vertex in the graph, in which case it is a leaf.
- 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.

Example: Terminologies

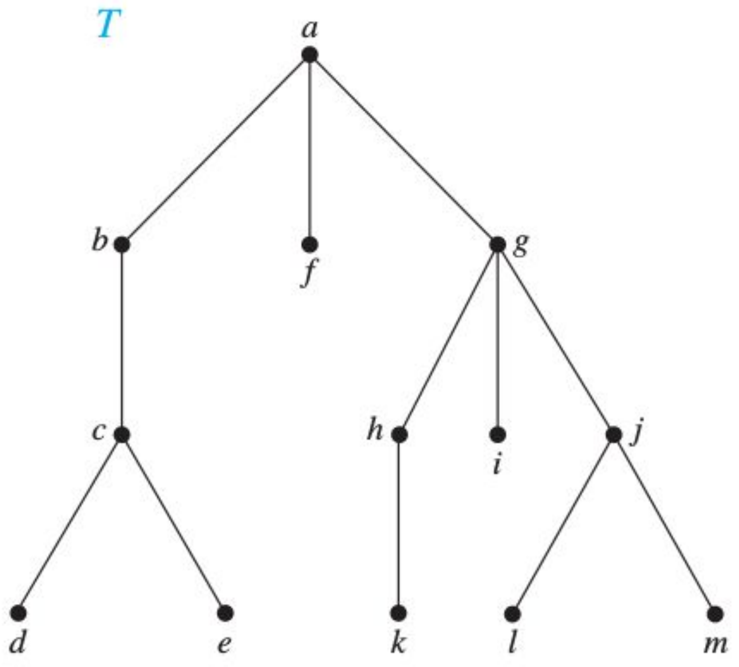


FIGURE 5 A Rooted Tree T.

In the rooted tree T (with root a) shown in Figure 5, 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.

Example: Terminologies

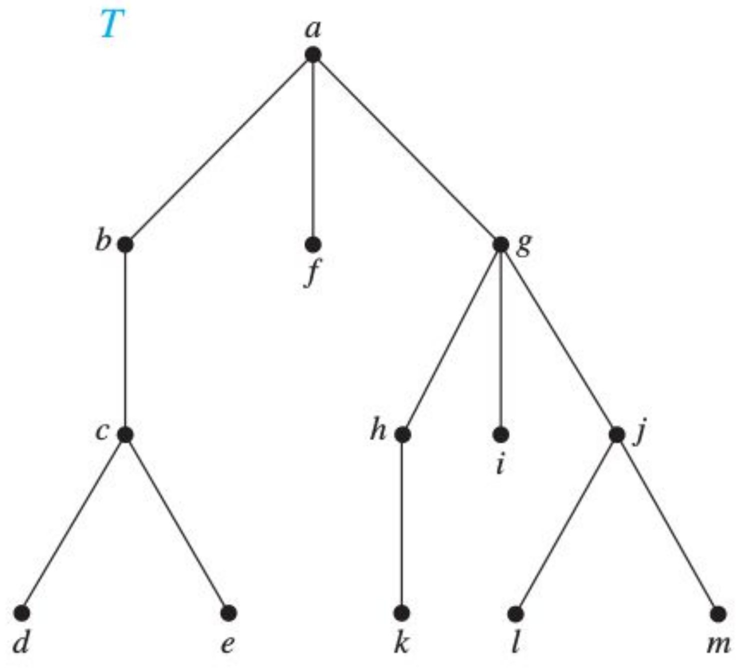


FIGURE 5 A Rooted Tree *T*.

In the rooted tree T (with root a) shown in Figure 5, 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.

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.

Example: Terminologies

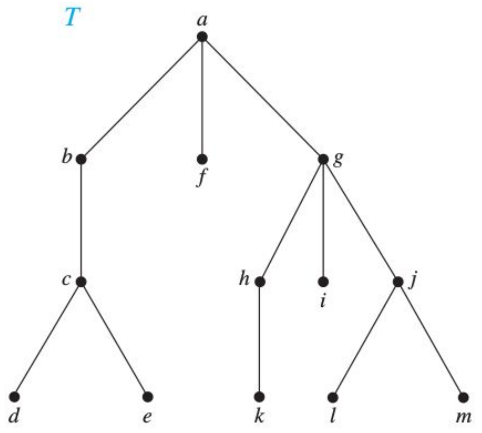


FIGURE 5 A Rooted Tree T.

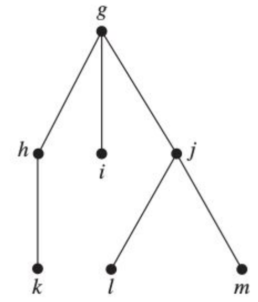


FIGURE 6 The Subtree Rooted at g.

What is the subtree rooted at g?

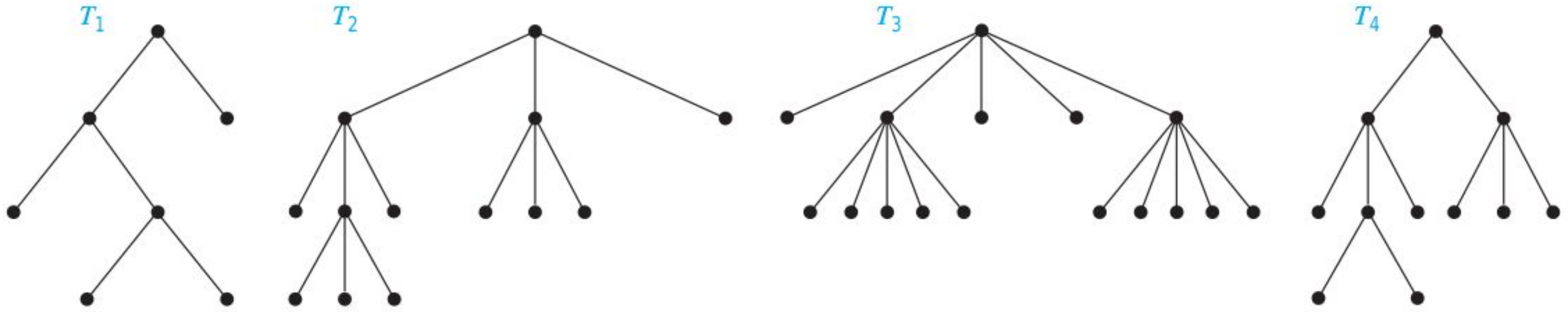
The subtree rooted at g is shown in Figure 6.

M-ary Tree

- 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.

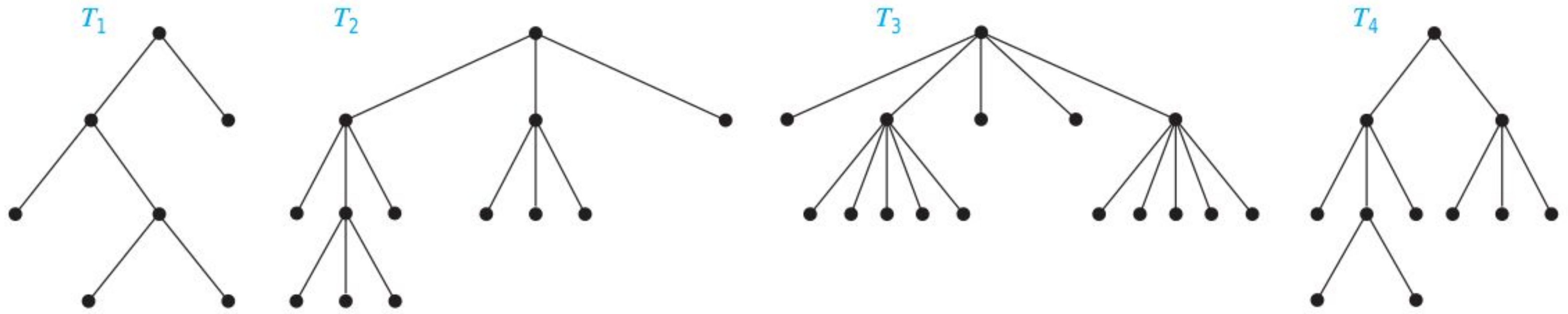
Examples: m-ary tree

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



Examples: m-ary tree

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



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.

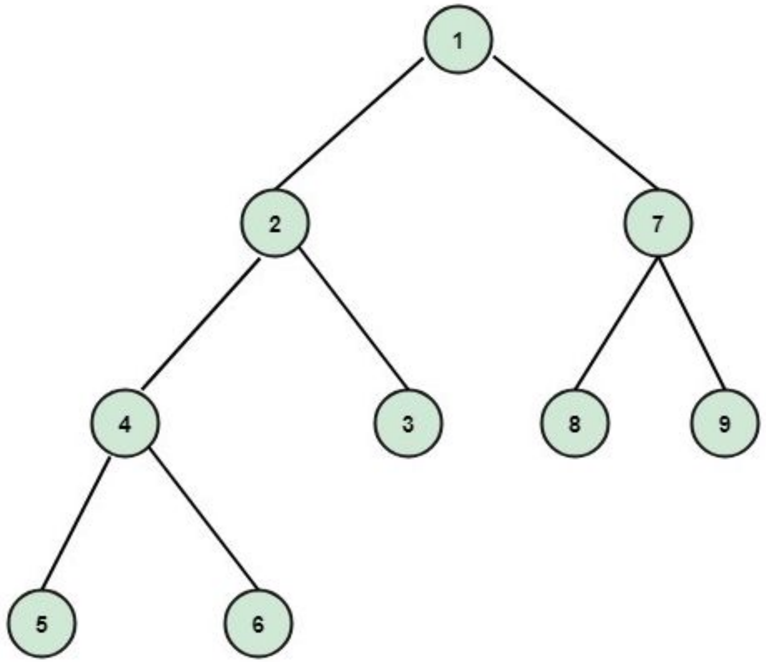
Ordered Rooted Trees

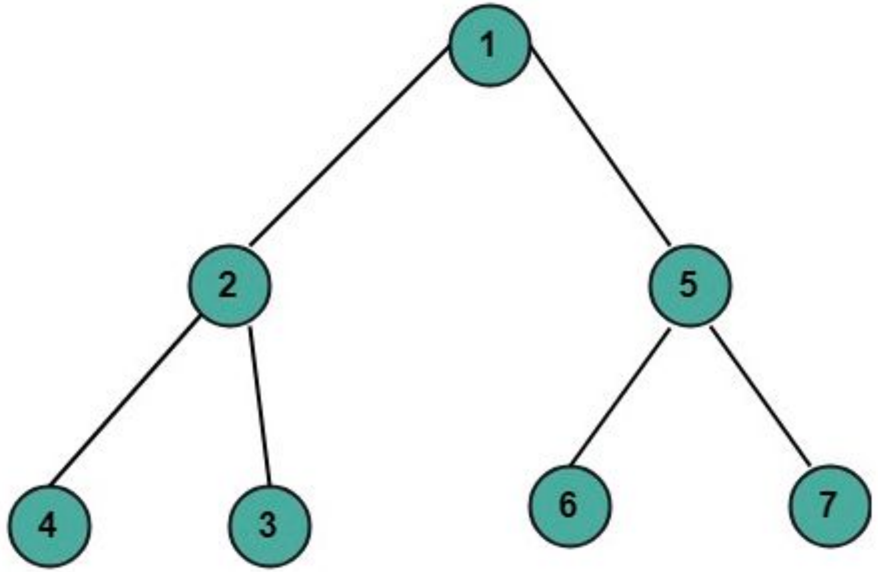
- An ordered rooted tree is a rooted tree where the children of each internal vertex are ordered. Ordered rooted trees are drawn so that the children of each internal vertex are shown in order from left to right.
- In an ordered binary tree (usually called just a binary tree), if an internal vertex has 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.

Other Trees

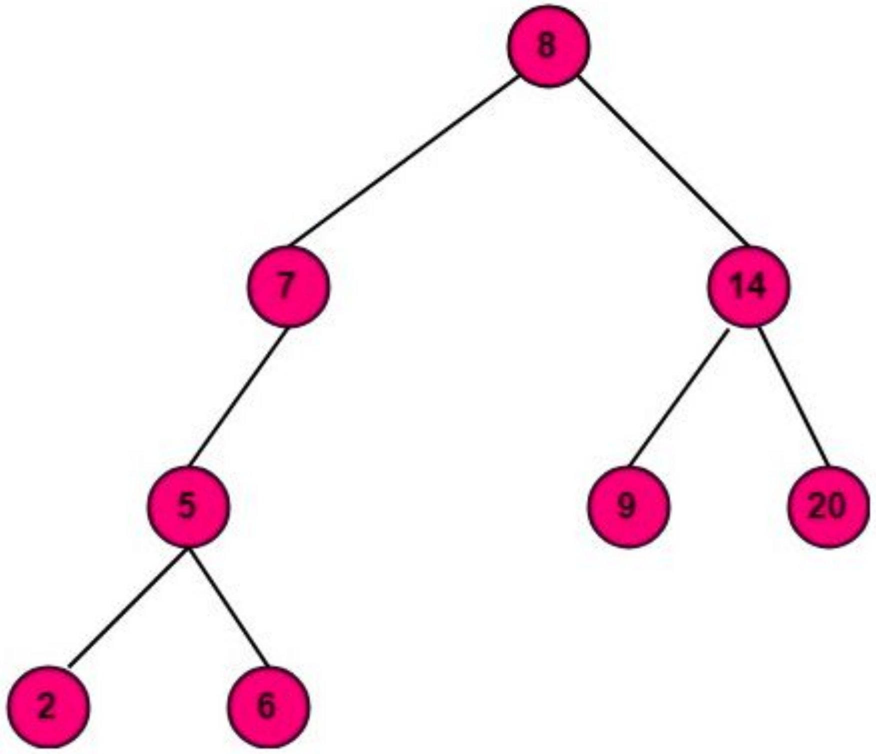
- Complete binary tree is a binary tree if it is all levels, except possibly the last, have the maximum number of possible nodes as for left as possible.
- Full binary tree is a binary tree in which all the leaves are on the same level and every non-leaf node has two children.
- Binary search trees have the property that the node to the left contains a smaller value than the node pointing to it and the node to the right contains a larger value than the node pointing to it.

Examples:





Examples:



Applications of trees

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

The English mathematician Arthur Cayley discovered trees in 1857 when he was trying to enumerate the isomers of compounds of the form C_ n H_2n+2 , which are called saturated hydrocarbons.

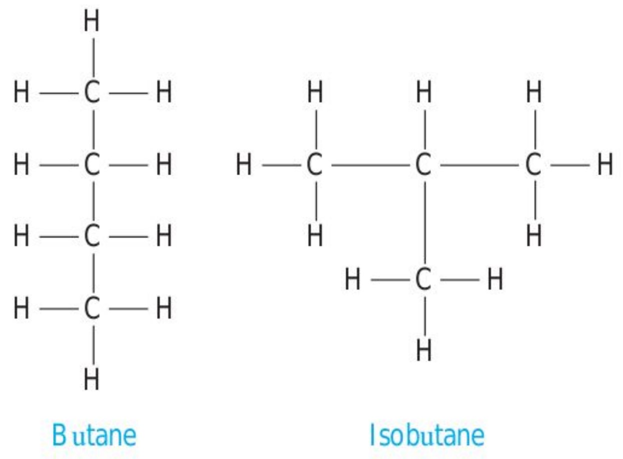


FIGURE 9 **The Two Isomers of Butane.**

Applications of trees

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.

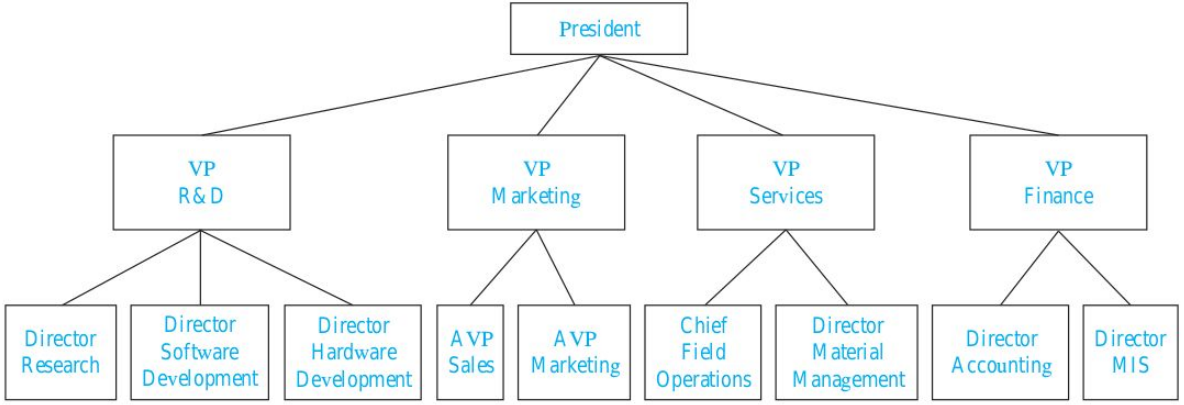


FIGURE 10 **An Organizational Tree for a Computer Company.**

Applications of trees

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.

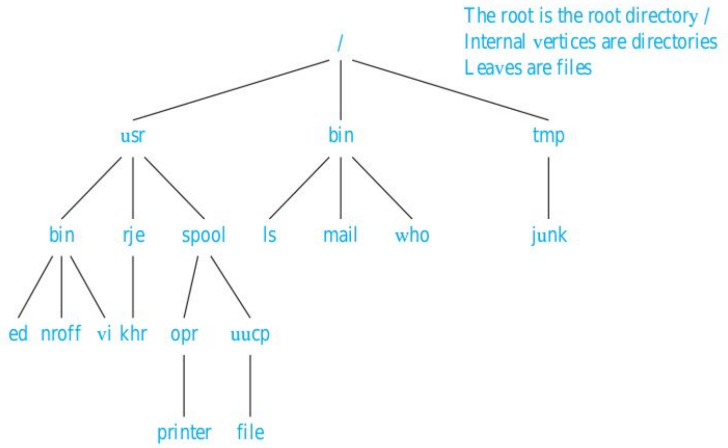


FIGURE 11 A Computer File System.

Tree Properties

Theorem 2: A tree with n vertices has $n - 1$ edges.

Tree Properties

Theorem 2: A tree with n vertices has $n - 1$ edges.

Proof: We will use mathematical induction to prove this theorem. Note that for all the trees here we can choose a root and consider the tree rooted.

BASIS STEP: When $n = 1$, a tree with $n = 1$ vertex has no edges. It follows that the theorem is true for $n = 1$.

INDUCTIVE STEP: The inductive hypothesis states that every tree with k vertices has $k - 1$ edges, where k is a positive integer.

Tree Properties

Theorem 2: A tree with n vertices has n − 1 edges.

Proof: INDUCTIVE STEP: The inductive hypothesis states that every tree with k vertices has k − 1 edges, where k is a positive integer. Suppose that a tree T has k + 1 vertices and that v is a leaf of T (which must exist because the tree is finite), and let w be the parent of v. Removing from T the vertex v and the edge connecting w to v produces a tree T with k vertices, because the resulting graph is still connected and has no simple circuits. By the inductive hypothesis, T has k − 1 edges. It follows that T has k edges because it has one more edge than T , the edge connecting v and w. This completes the inductive step.

Tree Properties

Theorem 3: A finite tree with more than one vertex has at least one vertex of degree one.

Proof: A tree is connected so there are no vertices of degree zero. Suppose for a contradiction that there are v vertices and $v - 1$ have degree at least two. Then the sum of the degrees of the vertices is at least $1 + 2(v - 1) = 2v - 1$, so the number of edges (which is always one half the sum of the degrees) is at least $v - 1/2$. This is impossible as a tree has exactly $v - 1$ edges.

Revisit: Tree Properties

Theorem 1: An undirected graph is a tree if and only if there is a unique simple path between any two of its vertices.

Theorem 2: A tree with n vertices has n − 1 edges.

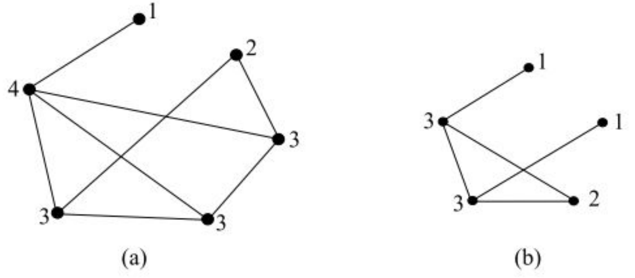
Theorem 3: A finite tree with more than one vertex has at least one vertex of degree one.

Degree sequence of a Tree

- A degree sequence lists out the degrees (number of edges incident to the vertex) of all the vertices in a graph in non-increasing order.
- Let di , 1 ≤ i ≤ n, be the degrees of the vertices vi of a graph in any order. The sequence [di]^n_1 is called the degree sequence of the graph.

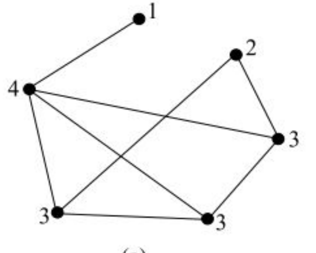
Degree sequence of a Tree

- A degree sequence lists out the degrees (number of edges incident to the vertex) of all the vertices in a graph in non-increasing order.
- Let di , 1 ≤ i ≤ n, be the degrees of the vertices vi of a graph in any order. The sequence [di]^n_1 is called the degree sequence of the graph.
- Examples:

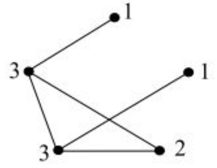


Degree sequence of a Tree

- A degree sequence lists out the degrees (number of edges incident to the vertex) of all the vertices in a graph in non-increasing order.
- Let di , 1 ≤ i ≤ n, be the degrees of the vertices vi of a graph in any order. The sequence [di]^n_1 is called the degree sequence of the graph.
- Examples:



(a)

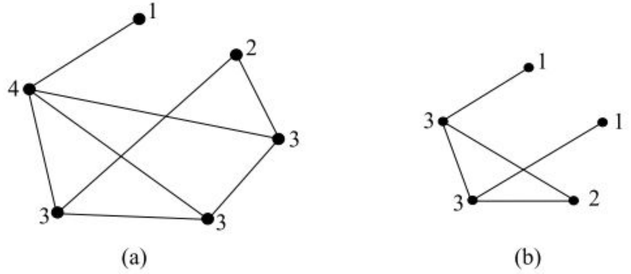


(b)

(a) {4,3,3,3,2,1} (b) {3,3,2,1,1}

Degree sequence of a Tree

- A degree sequence lists out the degrees (number of edges incident to the vertex) of all the vertices in a graph in non-increasing order.
- Let di , 1 ≤ i ≤ n, be the degrees of the vertices vi of a graph in any order. The sequence [di]^n_1 is called the degree sequence of the graph.
- Examples:



(a) {4,3,3,3,2,1} (b) {3,3,2,1,1}

- Trees (on n vertices) have n − 1 edges ⇒ Degree sum is 2n − 2
- Theorem 4: Positive integers d1, d2, . . . , dn are degrees of a tree ⇔ di = 2n − 2

Spanning Tree

- Let G be a simple graph. A spanning tree of G is a subgraph of G that is a tree containing every vertex of G.
- A simple graph is connected if and only if it has a spanning tree.

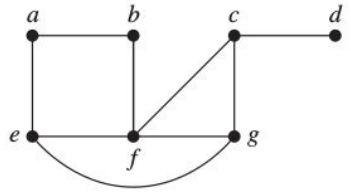


FIGURE 2 The Simple Graph G.

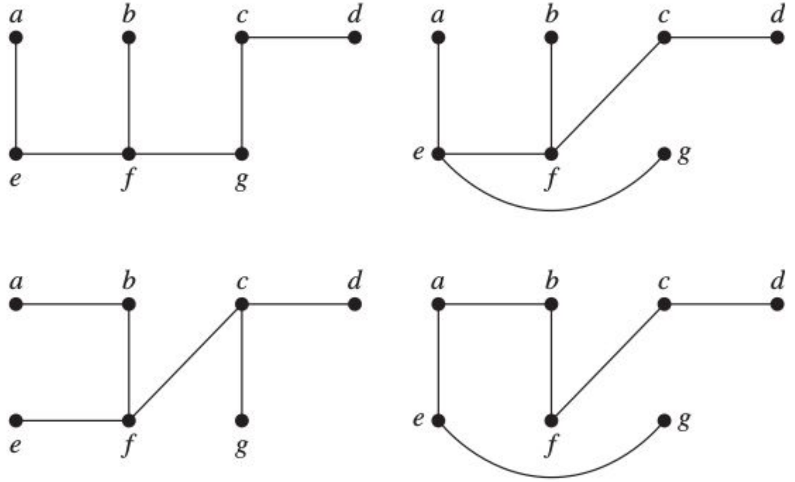


FIGURE 4 Spanning Trees of G.

Spanning Tree

- Let G be a simple graph. A spanning tree of G is a subgraph of G that is a tree containing every vertex of G.
- A simple graph is connected if and only if it has a spanning tree.

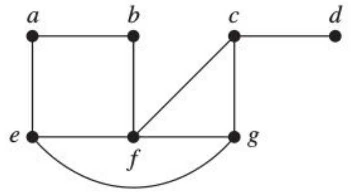


FIGURE 2 The Simple Graph *G*.

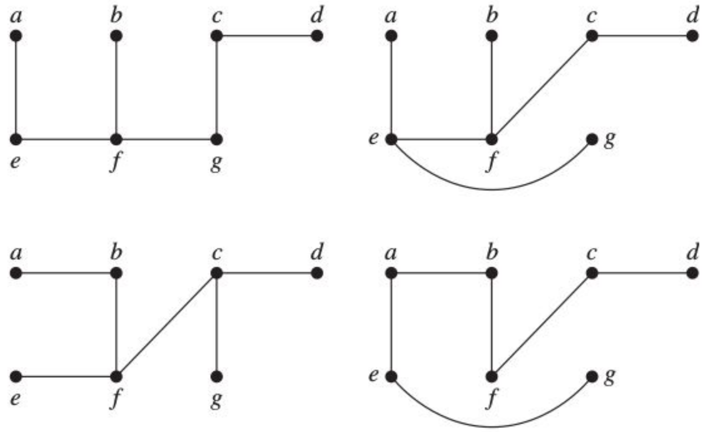


FIGURE 4 Spanning Trees of *G*.

A minimum spanning tree in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges.

References

[1] Chapter 10, 11 [KR]
[2] Chapter 6 [BDS]
[3] Chapter 1, 2 [RD]

Note: All images unless cited are from one of the above sources