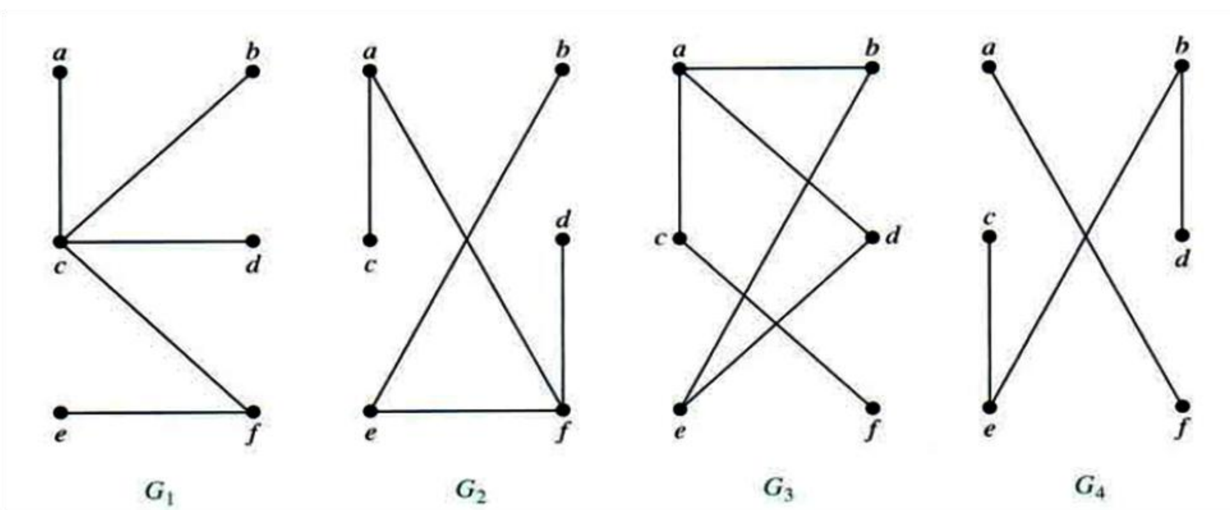
3.7

Trees

We have seen how graphs can be used to model and solve many problems. In this module we will focus on a particular type of graph called a **tree.** It is so named because such a graph resembles a tree. For example, family trees use vertices to represent the members of a family and edges to represent parent-child relationships.

Tree: A **tree** is a connected undirected graph with no simple circuits.
A tree does not contain multiple edges or loops, because it does not contain simple circuits. Therefore, a tree must be a simple graph.

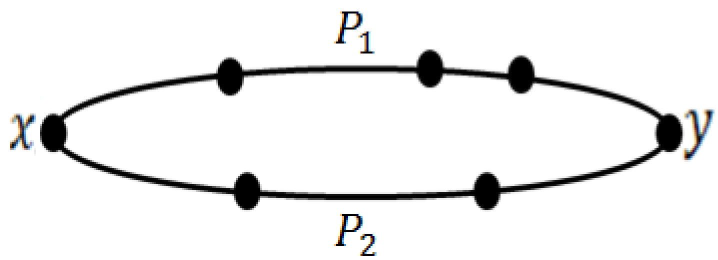
Example 1: Which of the following graphs shown below are tree?



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

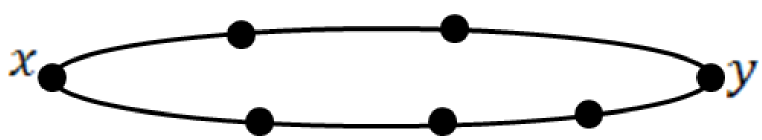
Theorem 1: An undirected graph is a tree if and only if there is a unique simple path between any two of its vertices.

Proof: Let T be an undirected graph.
Suppose that T is a tree. Let x and y be any two vertices. Because T is connected, there is a simple path between x and y.



To show that this path is unique, assume that there is another simple path between x and y. Then the path formed by combining the first path from x to y followed by the path from y to x(obtained by reversing the order of the second path from x to y) would form a simple circuit. This is a contradiction. Hence, there is a unique simple path between any two vertices of a tree.

Conversely, suppose that there is a unique simple path between any two vertices in the graph T. Therefore, T is connected. Assume that T has a simple circuit containing two vertices x and y. Then there would be two simple paths

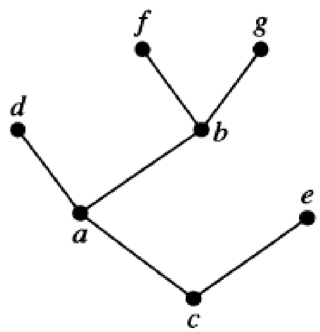


between x and y- a contradiction. This shows that T has no simple circuits. Thus, T is connected and has no simple circuits. Therefore, T is a tree.
Hence the theorem.

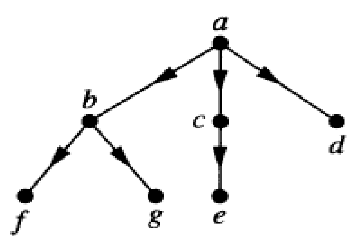
In many applications of trees, a particular vertex of a tree is designated as the *root***.** Because there is a unique path from the root to each vertex of the graph, we direct each edge away from the root. Once, we specify a root, we can assign a direction to each edge. Thus, a tree together with its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

Rooted tree: A **rooted tree** is a tree in which one vertex is designated as the root and every edge is directed away from the root.
We can change any undirected tree into a rooted tree by choosing any vertex as the root.

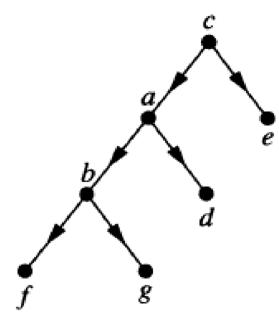
We usually draw a rooted tree with its root at the top of the graph. The rooted treed formed by designating a as the root and c as the root respectively in the tree T are given below:



T



Rooted tree with a as a root



Rooted tree with c as a root

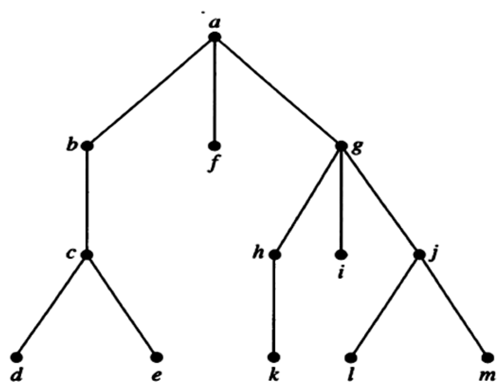
Note:

i. Different choices of the root produces different rooted trees.
ii. The arrows indicating the directions of the edges in a rooted tree can be omitted, because the choice of the root determines the direction of the edges.

Terminology: The terminology for trees has botanical and genealogical origins. Suppose, that T is a rooted tree. If v is a vertex in a tree T other than the root, then 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 **ancesestors** of a vertex v other than the root are the vertices in the path from the root to v, excluding v itself and including the root, $i.e$, its parent, its parent's parent, and so on, until the root is reached. The **descendents** of a vertex v are those vertices that have v as an ancestor. A vertex of a 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, its descendents and all edges incident with these descendents

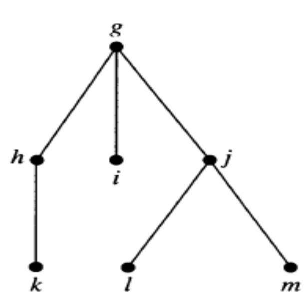
For the rooted tree T given below:



A Rooted Tree T.

i. The root of the rooted tree T is a vertex a.
ii. The parent of c is b
iii. The children of g are h, i and j
iv. The siblings of h are i and j
v. The ancestors of e are c, b and a
vi. The descendents of b are c, d and e
vii. The internal vertices of T are a, b, g, c, h and j.
viii. The leaves of T are d, e, f, i, k, l and m

The subtree rooted at g is shown below

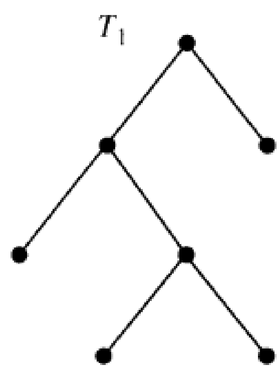


Rooted trees with the property that all the internal vertices have the same number of children are used in many applications. Such trees are used to study the problems related to searching, sorting and coding.

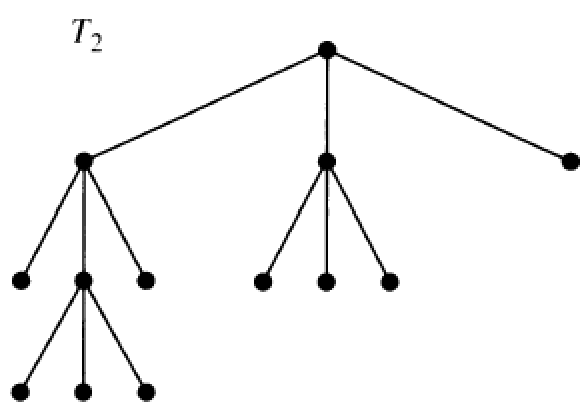
$\boldsymbol{m}$ **−ary, full** $\boldsymbol{m}$ **−ary and binary trees:** A rooted tree is called an $\boldsymbol{m}$ **−ary tree ,** if every internal vertex has no more than m children . A rooted tree is called a **full** $\boldsymbol{m}$ **−ary tree**, if every internal vertex has exactly m children. An $\boldsymbol{m}$ **−ary tree** with $m = 2$ is called a **binary tree.**

A **complete** $\boldsymbol{m - ary}$ **tree** is a full $m - ary$ tree **,** where every leaf is at the same level**.**

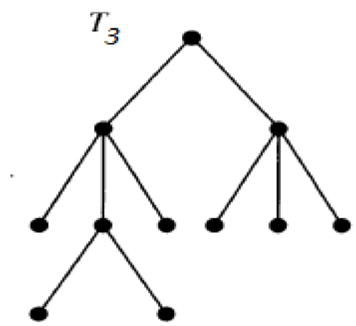
For example:



T_1 is a full binary tree, because each of its internal vertices has exactly two children.



T_2 is a full 3 —ary tree, because each of its internal vertices has exactly three children.

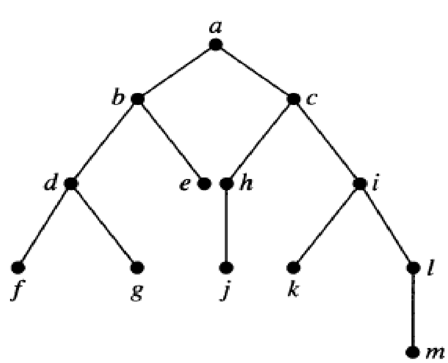


T_3 is not a full m —ary tree for any m, because some of its internal vertices have two children and others have three children. It is a 3 —ary tree, because each internal vertex has no more than three children.

Ordered root tree: An **Ordered tree** is a rooted where the children of each internal vertex are ordered.
Ordered root trees are drawn such that the children of each internal vertex are shown in the order from left to right.

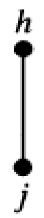
In an ordered binary tree(usually called just a binary tree), if an internal vertex has two children, the first child is called the **left child** and the second child is called the **right child**. The tree rooted at the left child of a vertex v is called the **left subtree** of v and the tree rooted at the right child of a vertex v is called the **right subtree** of v

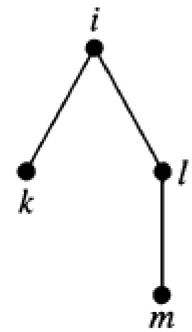
For example in the tree T shown below, the left child of d is f and
right child is g



A binary tree T

The following are the left subtree of the vertex c and the right subtree of the vertex c respectively:





Left subtree of the vertex c Right subtree of the vertex c

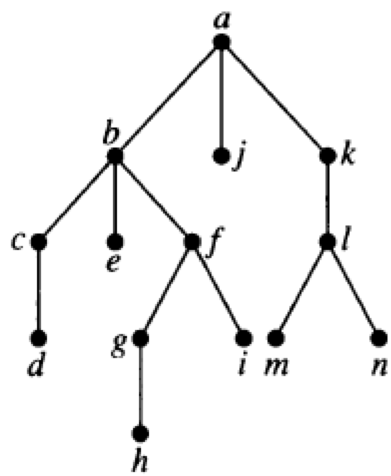
The following are the concepts related to the *balanced tree*.

Level of a vertex: The level of a vertex v in a rooted tree is the length of the unique path from the root to the vertex v. The level of the root is defined to be zero.

The height of the rooted tree: The height of a rooted tree is the maximum of the levels of the vertices. That is, the height of a rooted tree is the length of the longest path from the root to any vertex.

A rooted m −ary tree of height h **is balanced** if all its leaves are at levels h or $h - 1$

Example 2: Find the level of each vertex in the rooted tree given below. Find the height of the tree

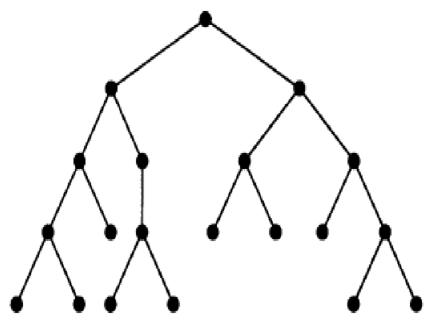


Solution:

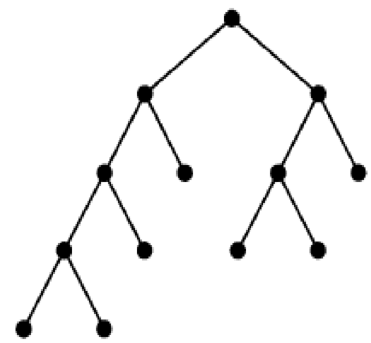
Vertex	Level
a (root)	0
b, j, k	1
c, e, f, l	2
d, g, i, m, n	3
h	4

The height of the tree is 4, because the largest level of any vertex is 4.

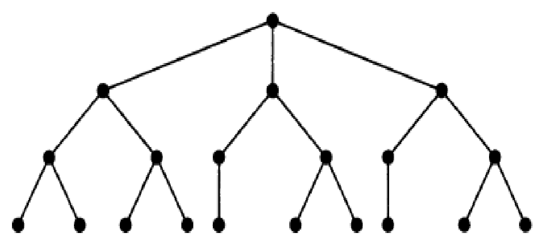
Example 3: which of the following rooted trees shown below are balanced



T_1



T_2



T_3

Solution: The height of T_1 is 4. Notice that all its leaves are at levels 3 and 4. Therefore, T_1 is a balanced tree.

The height of T_2 is 4. Notice, that all its leaves are at levels 2,3 and 4. Therfore, T_2 is not a balanced tree.

The height of T_3 is 3 . Notice that all its leaves are at level 3.Therefore, T_3 is a balanced tree

Properties of Trees: We often need results related to the number of edges and vertices in various types of trees.

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

Proof: We furnish proof by mathematical induction. Note that for any tree, we can choose a root and consider the rooted tree.

Basis Step: When $n = 1$, a tree with one vertex has no edges. Thus, The result is true for $n = 1$.

Inductive Step: The induction hypothesis is: Every tree with k vertices has $k - 1$ edges, where k is a positive integer. Let T be a tree with $k + 1$ vertices. Let v be a leaf of T(this is possible since the tree is finite). Let w be the parent of v.We remove the vertex v and the edge connecting w. The resulting graph is a tree T' with k vertices. By induction hypothesis T' has $k - 1$ edges. It follows that T has k edges because it has one more edge than T' ($i.e$ the edge connecting v and). By mathematical induction the result follows.

The following theorem gives the number of vertices in a full m –ary tree with a specified number of internal vertices. Throughout, the discussion n denotes the number of vertices in a tree.

Theorem 3: A full $\boldsymbol{m}$ –ary tree with $\boldsymbol{i}$ internal vertices contains $\boldsymbol{n = mi + 1}$ vertices.

Proof: Notice that every vertex, except the root, is the child of all internal vertices. Since each of the i internal vertices has exactly m children, there are mi vertices in the tree other than the root. Therefore, the tree has $n = mi + 1$ vertices. Hence the theorem

Let T be a full m –ary tree with n vertices.Let i be the number of internal vertices and l be the number of leaves in T. The following theorem gives the relationships among m, n, i and l.

Theorem 4: A full $\boldsymbol{m}$ –ary tree with $\boldsymbol{n}$ vertices has $\boldsymbol{i = \frac{n-1}{m}}$ internal vertices and $\boldsymbol{l = \frac{(m-1)n+1}{m}}$ vertices.

Proof: Let n, i and l respectively denote the number of vertices, the number of internal vertices and the number of leaves of an m –ary tree T. By Theorem 1 , $n = mi + 1$. Note that each vertex is either a leaf or an internal vertex. Therefore, $n = l + i$.Solving for i in $n = mi + 1$, we get $i = \frac{n-1}{m}$. Substituting the expression for i in the equation $n = l + i$ we obtain

$$l = n - i = n - \frac{(n-1)}{m} = \frac{(m-1)n+1}{m}$$

Corollary 1: A full $\boldsymbol{m}$ −ary tree with

(i) i Internal vertices have $n = mi + 1$ vertices and $l = (m-1)i + 1$ leaves.

(ii) l leaves have $n = \frac{ml-1}{m-1}$ vertices and $i = \frac{l-1}{m-1}$ internal vertices.

Example 4: 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 $\mathbf{100}$ people who read it but did not send it out? How many people sent out the letter?

Solution: The chain letter can be represented by a full 4 −ary tree, $i.e.\ m = 4$. The internal vertices (i) correspond to people who sent out the letter and the leaves (l) correspond to people who did not send it out.

Given that 100 people did not send out the letter, therefore $l = 100$ .The number of people who have seen the letter n and

$$n = \frac{ml-1}{m-1} = \frac{4.100-1}{4-1} = 133$$

The number of internal vertices $i = n - l = 133 - 100 = 33$ .Thus 33 people sent out the letter.

The following result estimates the number of leaves in an m −ary tree.

Theorem 5: There are atmost $\boldsymbol{m^h}$ leaves in an $\boldsymbol{m}$ −ary tree of height $\boldsymbol{h}$

Corollary 2: if an $\boldsymbol{m}$ −ary tree of height $\boldsymbol{h}$ has $\boldsymbol{l}$ leaves, then $\boldsymbol{h \geq \lceil log_m\, l \rceil}$.
If the $\boldsymbol{m}$ −ary tree is full and balanced, then $\boldsymbol{h = \lceil log_m\, l \rceil}$
(Here $\lceil x \rceil$ is the least integer grater than or equal to $\boldsymbol{x}$)

Note: A complete m −ary tree of height h has m^h leaves

Trees as Models: Trees are used as models in such diverse areas as Computer Science, Chemistry, Geology, Botany and Psychology.

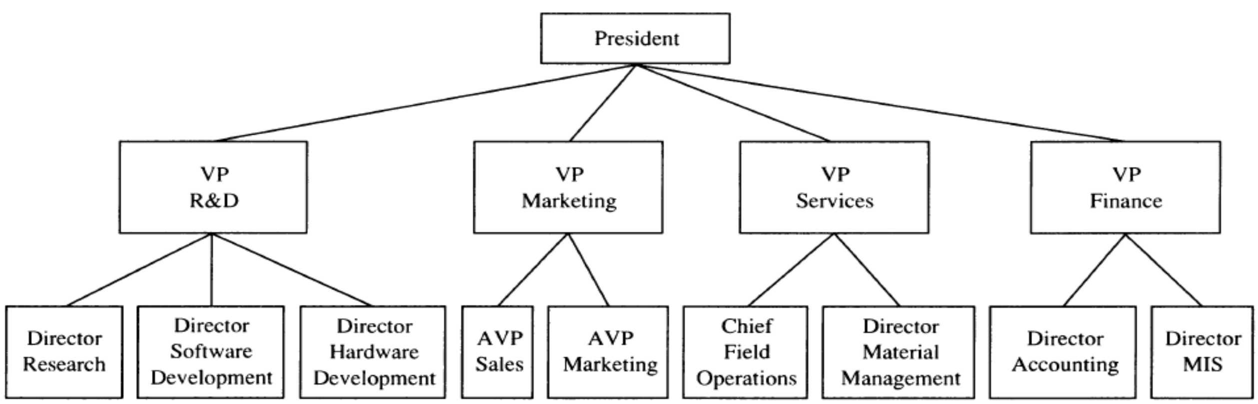
1. **Saturated Hydrocarbons and Trees**

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 enumente the isomers of compounds of the form C_nH_{2n+2}, which are called *Saturated Hydrocarbons*

2. **Representing Organizations**

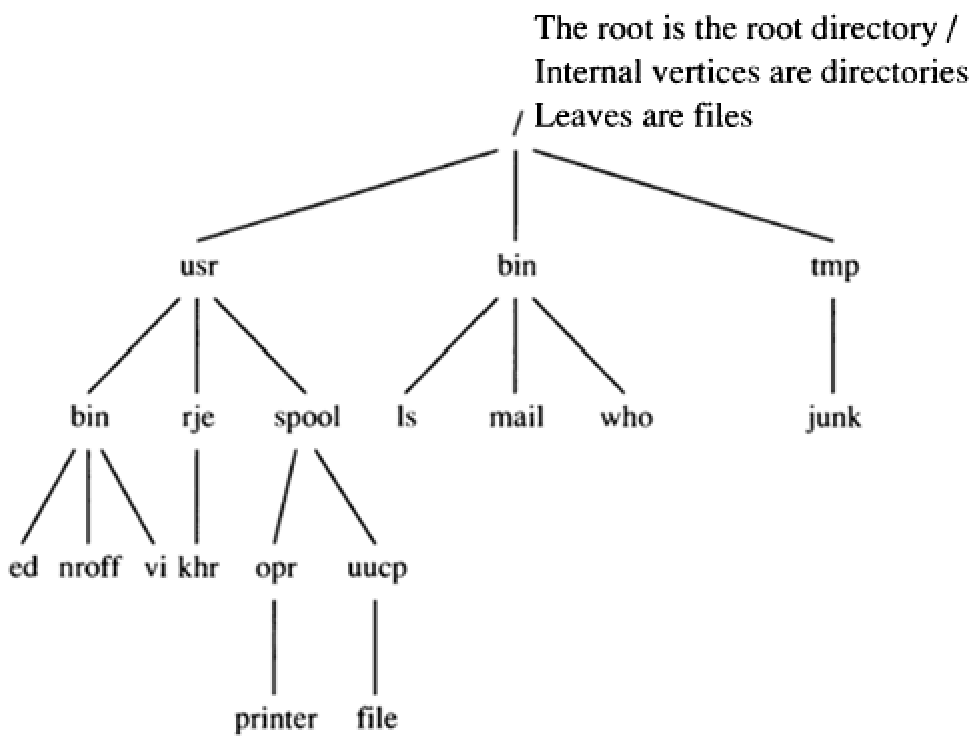
The structure of a large organization can be modeled using a rooted tree. Each vertex in this tree represents a position in the organization. An edge from one vertex to another vertex indicates that the person represented by the initial vertex is the direct boss of the person represented by the terminal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



An organizational tree for a computer company

3. **Complier File System**

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 the subdirectories, and the leaves represent ordinary files or empty directories. One such file system is given below:

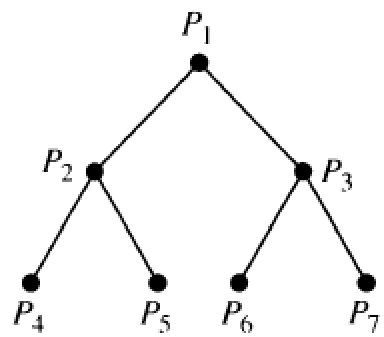


A computer File system

Note: The links to files where the same file may have more than one pathname can lead to circuits in computer file systems

4. Tree –Connected Parallel Processors

A **tree connected network** is an important way to interconnect the processors. The graph representing such a network is a complete binary tree. Such a network interconnects $n = 2^k - 1$ processors, where k is a positive integer. A processor represented by the vertex v that is not a root or a leaf has three two-way connections, one to the processor represented by the parent of v and two to the processor represented by the two children of v. The processor represented by the oot has two-way connections to the processors represented by its two children. A processor represented by a leaf u has a single two-way connections to the parent of u.



A Tree-Connected Network of Seven Processors

Illustration of the use of a tree –connected network for parallel computation

This is an illustration of the use of processors in the above figure to add eight numbers $x_1, x_2, \ldots, x_8$ using three steps.

In the first step, we add x_1 and x_2 using P_4, x_3 and x_4 using P_5, x_5 and x_6 using P_6 and x_7 and x_8 using P_7.

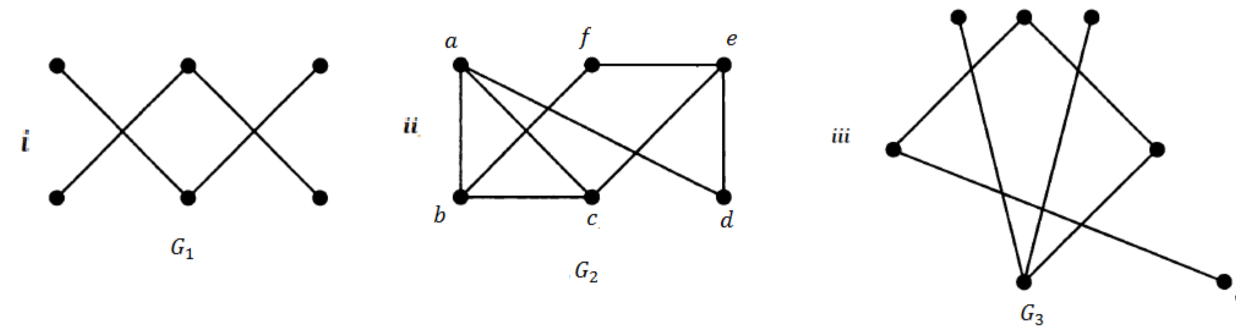
In the second step, we add $x_1 + x_2$ and $x_3 + x_4$ using P_2 and $x_5 + x_6$ and $x_7 + x_8$ using P_3.

In the third step, we add $x_1 + x_2 + x_3 + x_4$ and $x_5 + x_6 + x_7 + x_8$ using P_1

The three steps used to add eight numbers compares favorably to the seven steps required to add eight numbers serially, where the steps are the addition of one number to the sum of the previous numbers in the list.

P1:

Which of the following graphs shown below are trees?



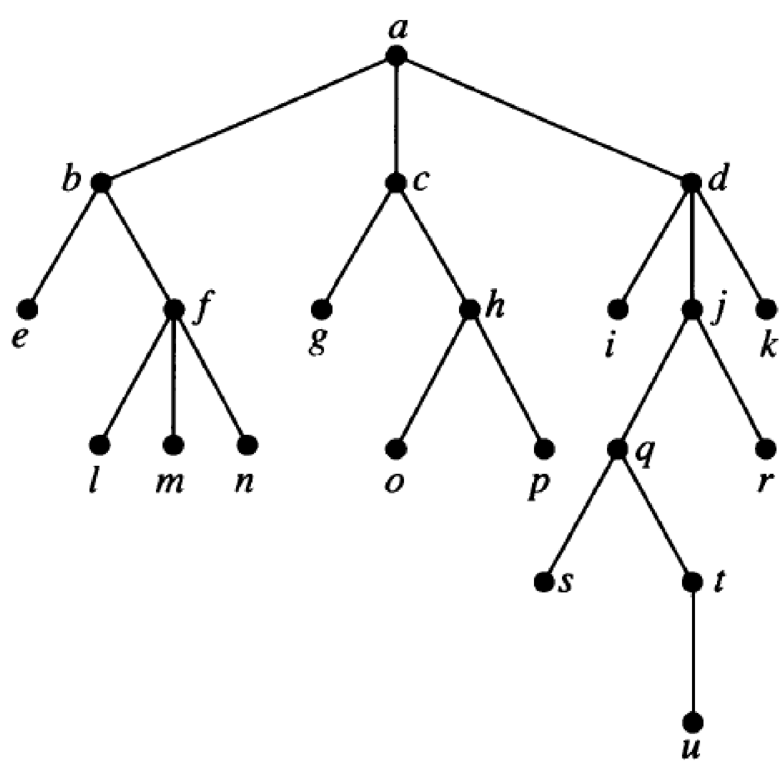
Solution:

Recall that a tree is a connected undirected graph with no simple circuits

i. The graph G_1 is not a tree, because it is not connected.
ii. The graph G_2 is not a tree , because it has a simple circuit a, b, c, a and also b, c, e, f, b
iii. The graph G_3 is a tree, because it is connected and has no simple circuits.

P2:

Answer the following questions for the rooted tree given below.



a. **Which vertex is the root?**
b. **Which vertices are internal?**
c. **Which vertices are leaves?**
d. **Which vertices are children of j?**
e. **Which vertex is the parent of h?**
f. **Which vertices are the siblings of l?**
g. **Which vertices are ancestors of m?**
h. **Which vertices are descendants of b?**

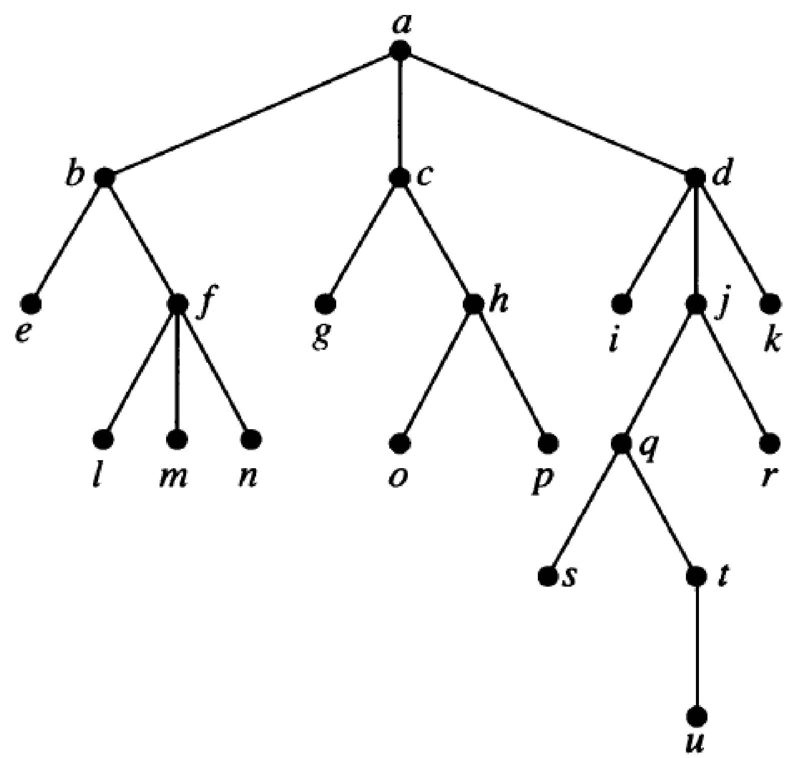
Solution:

a. Vertex a is the root
b. The vertices that have children are internal vertices. The internal vertices of this tree are a, b, c, d, f, h, j, q and t.

c. The vertices that have no children are leaves. The leaves are $e, g, i, k, l, m, n, o, p, r, s$ and u.
d. A vertex x is a child of a vertex y, if there is an edge from y to x.The children of j are q and r.
e. A vertex y is the parent of a vertex x, if there is a edge from y to x.The parent of the vertex h is c.
f. Vertices with the same parent are siblings. The siblings of the vertex l are m and n.
g. The ancestors of a vertex v are the vertices present in the path from the root to v, excluding v. The ancestors of the vertex m are f, b, a.
h. The descendants of a vertex v are those vertices that have v as an ancestor. The descendants of b are e, f, l, m and n.

P3:

Consider the following rooted tree.



a) Is the above rooted tree a full m-ary tree for some positive integer m

b) What is the level of each vertex of the above rooted tree.

c) What is the height of the above rooted tree.

Solution:

a) A rooted tree is an m-ary tree if every internal vertex has no more than m children. Notice that every internal vertex has no more than 3 children. Therefore, it is 3-ary tree.
A rooted tree is a full m –ary tree if every internal vertex has exactly m children. It is not a full m-ary tree for any positive integer m because some of its internal vertices have two children, the others have 3 children.

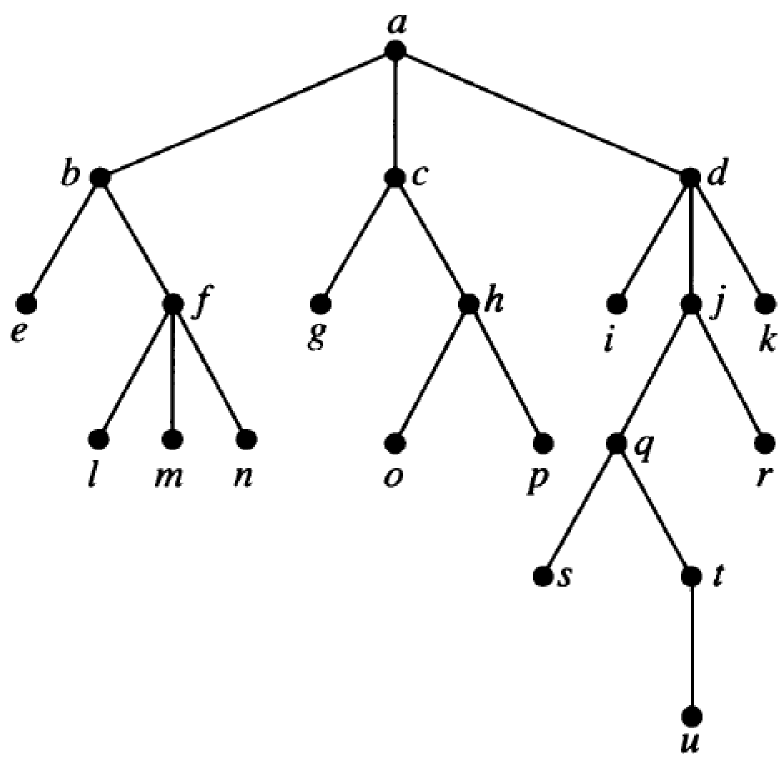
b)

vertices	level
a(root)	0
b, c, d	1
e, f, g, h, i, j, k	2
l, m, n, o, p, q, r	3
s, t	4
u	5

c) The largest level of any vertex is the height of the tree. Therefore, the height of the tree is 5.

P4:

Draw the subtree of the rooted tree given below rooted at (i) a (ii) c (iii) e.

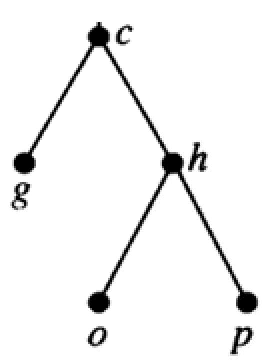


Solution:

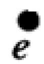
The subtree with v as its root is the subgraph of the tree consisting v and its descendents and all edges incident with these descendents.

(a) The sub tree rooted at a is the entire tree.

(b) The sub tree rooted at c is



(c) The sub tree rooted at e is



$i.e.$, e alone.

P5:

How many edges does a full binary tree with 1000 internal vertices have?

Solution:

We have a full binary tree, i.e., $m = 2$ and 1000 internal vertices, $i.e.$, $i = 1000$.

A full m-ary tree with i internal vertices has $n = mi + 1$ vertices.

Therefore, $n = 2 \times 1000 + 1 = 2001$ vertices.

A tree with n vertices has $e = n - 1$ edges. Therefore, $e = 2001 - 1 = 2000$ edges.

Thus, it has 2000 edges.

Note: If it has l leaves then $n = l + i$

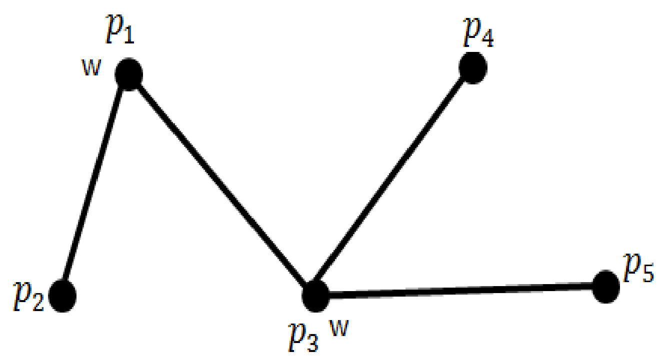
Therefore, $l = n - i = 2001 - 1000 = 1001$.Thus, it has 1001 leaves.

P6:

Suppose 1000 people enter a chess tournament. Use a rooted tree model of the tournament to determine the number of games must be played to determine a champion, if a player is eliminated after one loss and games are played until only one entrant has not lost. Assume that there are no ties.

Solution:

This can be modeled by a rooted tree, where the vertices are players and a game between two players is the edge between the corresponding vertices.



In the above example, there are 5 players $p_1, p_2, \ldots, p_5$. First p_1 and p_2 played and p_1 won. Now p_1 and p_3 played and p_3 won. It is now the turn of p_3 and p_4, and p_3 won. Finally p_3 and p_5 played and p_3 won. Thus the champion is p_3. The number of games played is $e = n - 1$ where n is the number of players.

Thus, the number of games played to determine a champion is the number of edges in a rooted with n vertices, $i.e.$, $e = n - 1$. Therefore, the number of games must be played to determine a champion is $1000 - 1 = 999$.

P7:

Either draw an m-ary tree with 84 leaves and height 3, where m is a positive integer, or show that no such tree exists.

Solution:

It is given height, $h = 3$ and $l = 84$.

An m-ary tree of height h has atmost m^h leaves. $i.e.$, $l \leq m^h \Rightarrow 84 \leq m^3$

This shows that $m \geq 5$.

By Theorem 4, a full m-ary tree with l leaves have $i = \frac{l-1}{m-1} = \frac{84-1}{m-1}$ leaves.

$i.e.$, since 83 is prime, we must have $m - 1 = 1$ or 83, $i.e.$, $m = 2$ or 84.

We take $m = 84$, because $m \geq 5$.

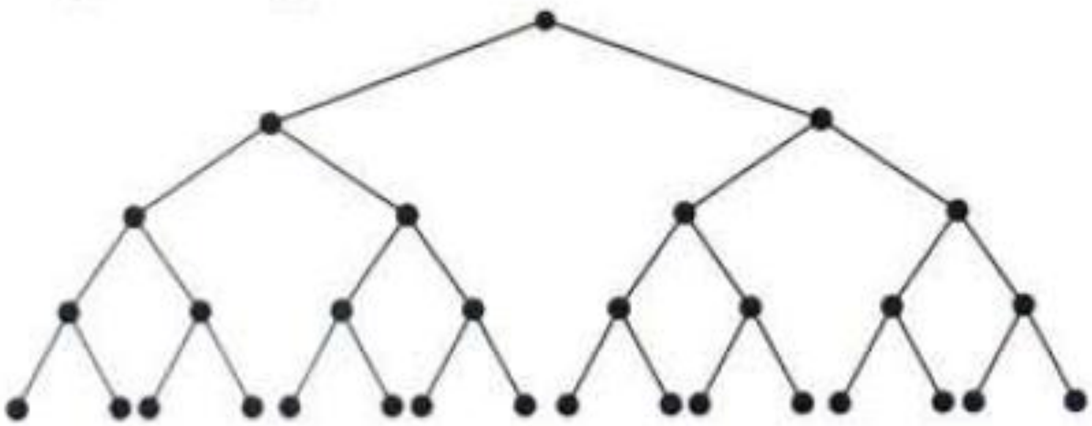
If $m = 84$, then $i = 1$ and its height is 1, which is a contradiction. Therefore, there is no such tree.

P8:

Construct a complete binary tree of height 4.

Solution:

The complete binary tree of height 4 is

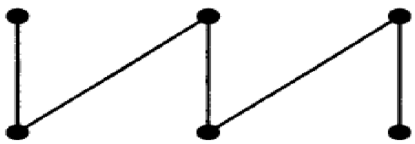


3.7. Trees

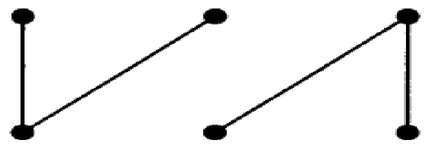
Exercise:

1. Which of the following graphs shown below are trees?

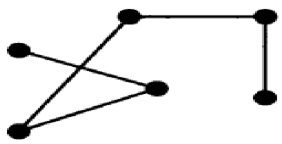
a)



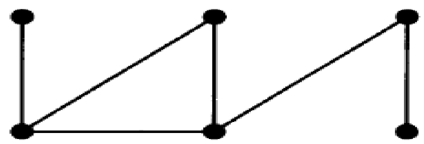
b)



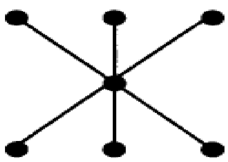
c)



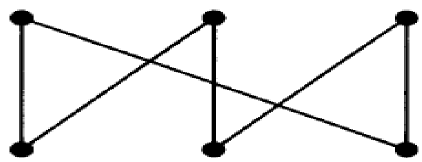
d)



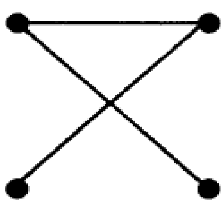
e)



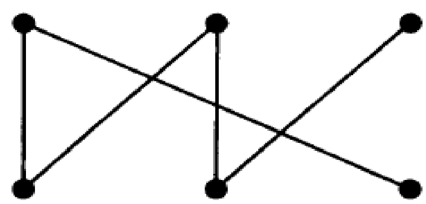
f)



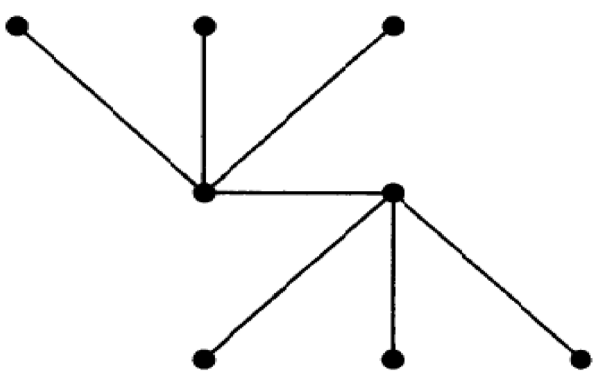
g)



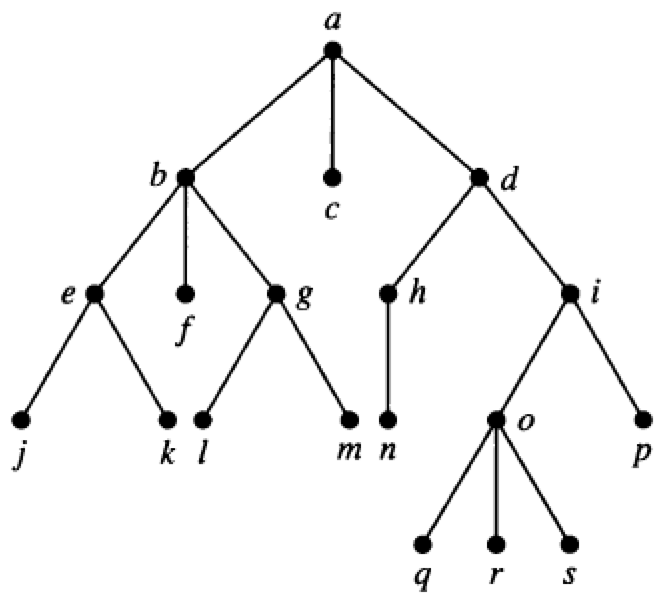
h)



i)



2. Consider the following rooted tree and answer the following questions?



a. Which vertex is the root?
b. Which vertices are internal?
c. Which vertices are leaves?
d. Which vertices are children of o?
e. Which vertex is the parent of h?
f. Which vertices are the siblings of l?
g. Which vertices are ancestors of m?
h. Which vertices are descendants of b?

3. Is the rooted tree in Exercise 2 a full m −ary tree for some positive integer m?

4. What is the level of each vertex of the rooted tree in Exercise 2?

5. Draw the subtree of the tree in Exercise 2 that is rooted at

a. a b. c c. e d. i

6. How many edges does a tree with 10,000 vertices have?

7. How many vertices does a full 5-ary tree with 100 internal vertices have?

8. How many leaves does a full 3-ary tree with 100 vertices have?

9. A chain letter starts when a person sends a letter to five others. Each person who receives the letter either sends it to five other people who have never received it or does not send it to anyone. Suppose that 10,000 people send out the letter before the chain ends and that no one receives more than one letter. How many people receive the letter, and how many do not send it out?