Date
Page

Chapter-4

* Tree :- undirected, connected, no simple

T.U. 2080

* Theorem :- An undirected graph is a tree iff there is a unique simple path between any two of its vertices.

* Proof :-
First, we assume that T be a tree. Then by definition, T is a connected graph with no simple circuit. As we know that there exists a simple path between any two vertices of a connected undirected graph. Thus 'T' must have a simple paths between any vertices x and y.

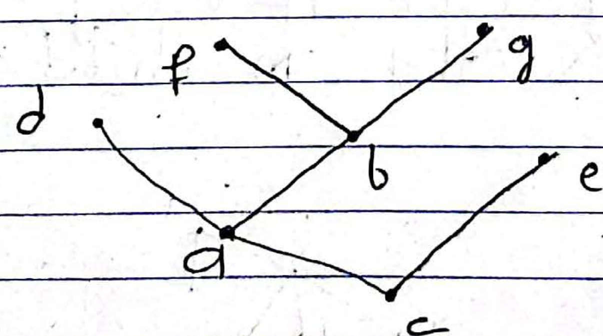
* Uniqueness :- Suppose the simple path between x and y is not unique, then there is a simple circuit between x and y. This contradicts that the graph T has no simple circuits. Hence there exists a unique simple path between any two vertices of the graph.

Date
Page

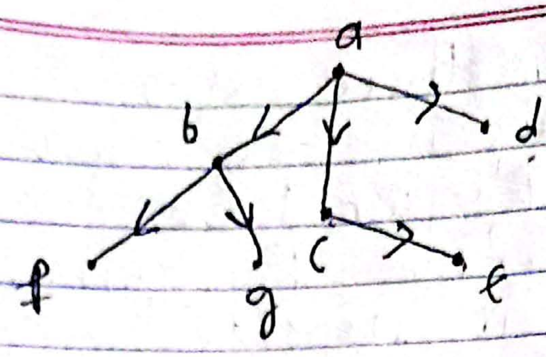
*/ Conversely, suppose there exists a unique simple path between any two vertices of the graph 'T' we have to show that graph T is a tree. For this we have to show that T is connected and T has no simple circuit. Since, there exists a unique simple path between any two vertices of graph T. So, T is connected.

Suppose T has a simple circuit that contains vertices x and y. Then, there would be simple path between x and y and simple path between y and x. This contradicts that T has a unique simple path. Hence, T has no simple circuit. Thus T is a tree.

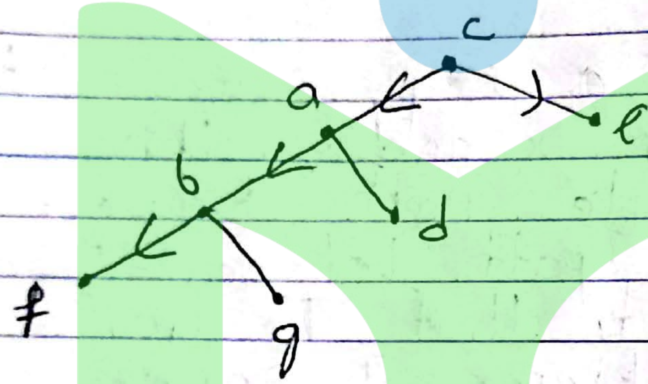
*/ Rooted Tree :- A rooted tree is a tree in which one vertex has been designated as the root and every edge is directed away from the root.



with root 'a'.



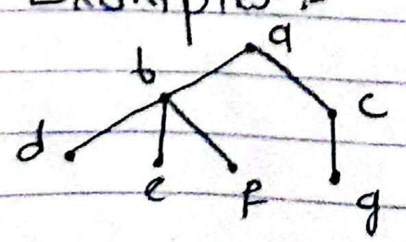
with root 'c'



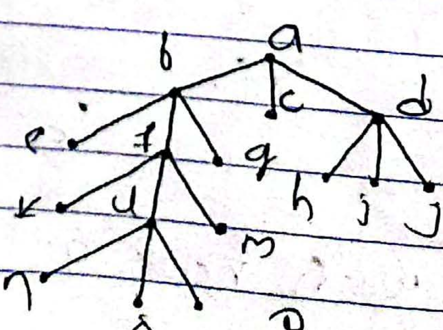
→ m-ary tree :- A rooted tree is called m-ary tree if every internal vertices has at most m children.

→ Full m-ary tree :- The tree is called a full m-ary tree if every internal vertices has exactly 'm' children.

* Examples :-

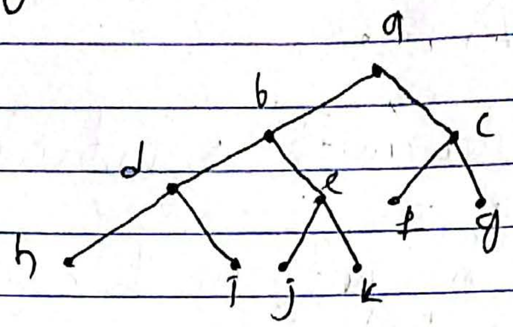
① 

3-ary tree



Date
Page

* Binary tree :- A tree having each vertex has exactly 2 children.



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

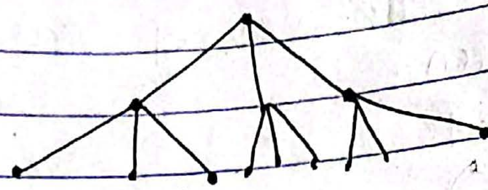
Proof :- We prove this theorem by using mathematical induction.

For $n = 1$, then the theorem is obvious. (i.e. no edges (zero edges))

Suppose this theorem is true for $n = k$ where k is a positive integer. Then when the number of vertices is k then the number of edges is $k-1$.

Now, we have to show that it is true for $n = k+1$.

For this suppose that T is a tree with number of vertices $n = k+1$. Let N be the leaf of T and w be its parents. Let T' be a tree by removing the vertex V and the edge containing w and V.

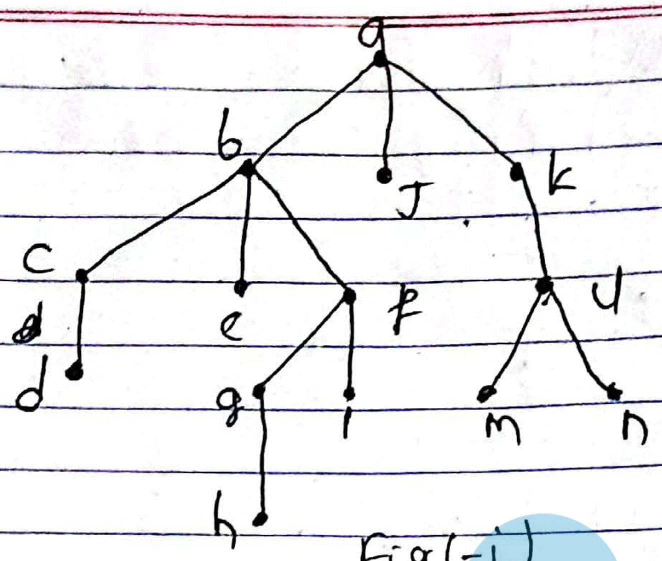


Then T is a tree with number of vertices $k = (k+1) - 1$.
So by induction hypothesis the number of edges in T' is $k-1$. Hence, the number of edges in T is $(k-1)+1 = k$.
Thus a tree with n vertices has $(n-1)$ edges.

Theorem :- A full m-ary tree with 'i' internal vertices contains $n = mi + 1$ vertices.

Proof :-
Every vertex except root is the child of an internal vertex. Because each of the i internal vertices has exactly m-children, there are mi vertices in the tree other than the root. Therefore the tree contains $n = mi + 1$ vertices.

Date
Page



Fig(-1)

*// Height of a rooted tree :- The height of a rooted tree is the maximum of levels of vertices.
In other words, the height of a rooted vertex is the length of the longest path from the root to any vertex.

*// Balanced rooted tree :- A rooted m-ary tree of height h is balanced if all leaves are at levels h or h-1.

*// Example :- Find the level of each vertices in the rooted tree shown in figure-1. What is the height of this tree.

⇒ Solution :-
The root a is at level zero.
The vertices b, j and k at level 1.

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

or, $n = \dfrac{lm + n - 1}{m}$

or, $nm = lm + n - 1$

or, $nm - n = lm - 1$

or, $n(m-1) = lm - 1$

$$\therefore \boxed{n = \frac{lm-1}{m-1}}$$

Again from (a) and (b), we got

$n = mi + 1 = l + i$

or, $mi - i = l - 1$

or, $(m-1)i = l - 1$

$$\therefore \boxed{i = \frac{l-1}{m-1}}$$

* Level of vertex:- The level of a vertex in a rooted tree is the level of the unique path from the root to this vertex.

The level of the root is zero.

Date ______
Page ______

or, $n = \frac{n-1}{m} + d$

or, $mn = n - 1 + md$

or, $d = n - mn + md$

or, $d - md = n(1-m)$

or, $d(1-m) = n(1-m)$

or, $d(m-1) = n(m-1)$

or, d

(ii) we have

$n = mi + 1$

and $n = i + d$

or, $i + d = mi + 1$

or, $d = mi - i + 1$

or, $d = (m-1)i + 1$

(iii) we have $n = mi + 1$

or, $i = \frac{(n-1)}{m}$

Also, we have $n = d + 1$

Putting the value of i, we get.

Date ____
Page ____

Theorem:- A full m-ary tree with

(i) n vertices has $i = \frac{n-1}{m}$ internal vertices and $l = \frac{[(m-1)n+1]}{m}$ leaves.

(ii) i internal vertices has $n = mi + 1$ vertices and $l = (m-1)i + 1$ leaves.

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

⇒ Proof:-

Let n, i and l represent the number of vertices, internal vertice and leaves, respectively.

we have, $n = mi + 1$ — (a)

Since, total number of vertices is the sum of internal vertices and leaves

$$n = i + l \quad \text{— (b)}$$

(i) we have $n = mi + 1$

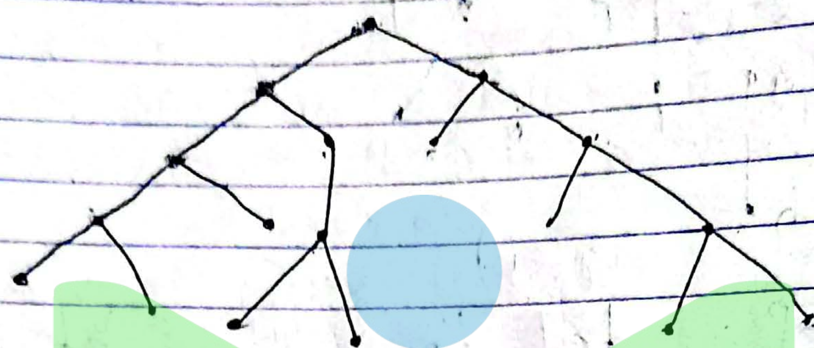
or, $n - 1 = mi$

$$\therefore i = \frac{n-1}{m}$$

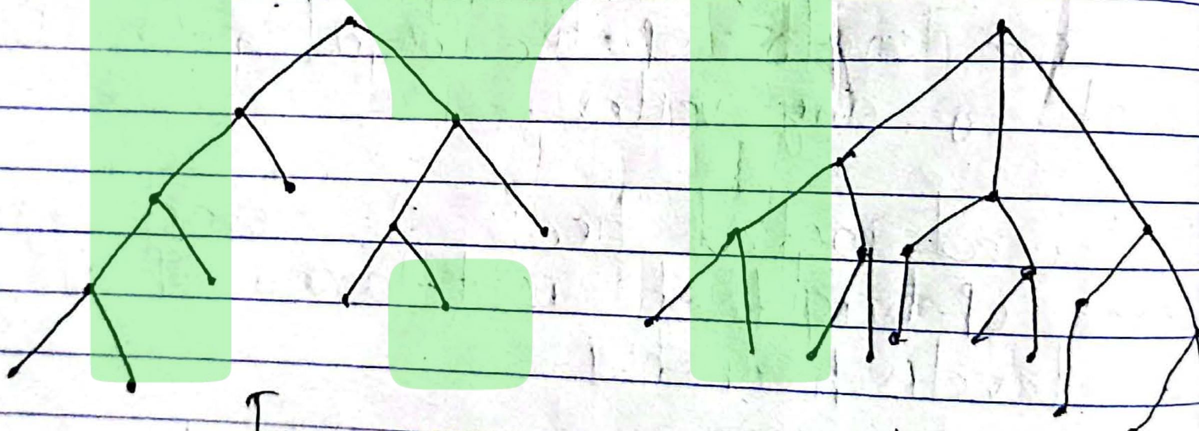
put the value of i in (b) we get.

Date
Page

Example: Which of the rooted tree shown is given figure below are balanced?



T1



T2

T3

⇒ Solution: Rooted tree T_1 is balanced because all its leaves are at height 3 and 4.

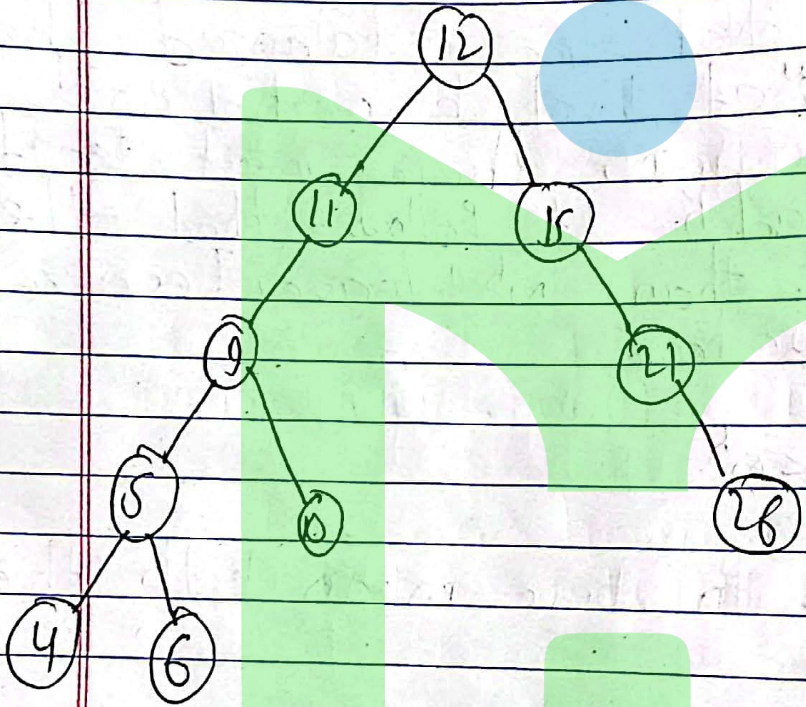
Date
Page

Binary search tree:-

→ Example (1):

Build a binary search tree by using numbers.

12, 15, 11, 9, 5, 21, 26, 4, 6, 10



→ Example (2):- From a binary search tree for the words mathematics, physics, Geography, zoology, meteorology, Geology, psychology and chemistry using alphabetical order.

→ Taking mathematics as the root of the tree. Since physics comes after mathematics

Date ______
Page ______

Since, h is an integer, we have

$\lceil \log_m l \rceil \le h$.

$\therefore h \ge \lceil \log_m l \rceil$

Now, suppose that tree is balanced. Then each leaf is at level h or h-1 and because height is h, there is at least one leaf at level h. It follow that there must be more than m^{h-1} leaves because $l \le m^h$ we have

$m^{h-1} \le l \le m^h$.

taking log with base m on both the sides we get

$\log_m m^{h-1} \le \log_m l \le \log_m m^h$

or, $h-1 < \log_m l \le h$ $(\because \log_m m = 1)$

Hence, $\lceil \log_m l \rceil = h$

Each of these subtrees has height less than or equal to $h-1$. So, by induction hypothesis, each of these rooted tree has at most m^{h-1} leaves.
Because there are at most m such subtree each with a maximum of m^{h-1} leaves. Therefore there are at most $m \times m^{h-1}$ leaves.
i.e. m^h leaves.
Hence, there are at most m^h leaves of a rooted tree of height h.

*// Corollary :- If an m-ary tree of height 'h' has l leaves, then $h \geq \lceil \log_m l \rceil$
If the m-ary tree is full and balanced, then $h = \lceil \log_m l \rceil$

=> Proof :- We know that $l \leq m^h$ (by previous theorem)

Taking log with base m both the sides we get

$\log_m l \leq \log_m m^h$

or, $\log_m l \leq h \log_m m$

or, $\log_m l \leq h \cdot 1$ [$\because \log_m m = 1$]

or, $\log_m l \leq h$

Date
Page

A bound for the number of leaves in an m-ary tree :-

Theorem :- There are at most m^h leaves in an m-ary tree of height h.

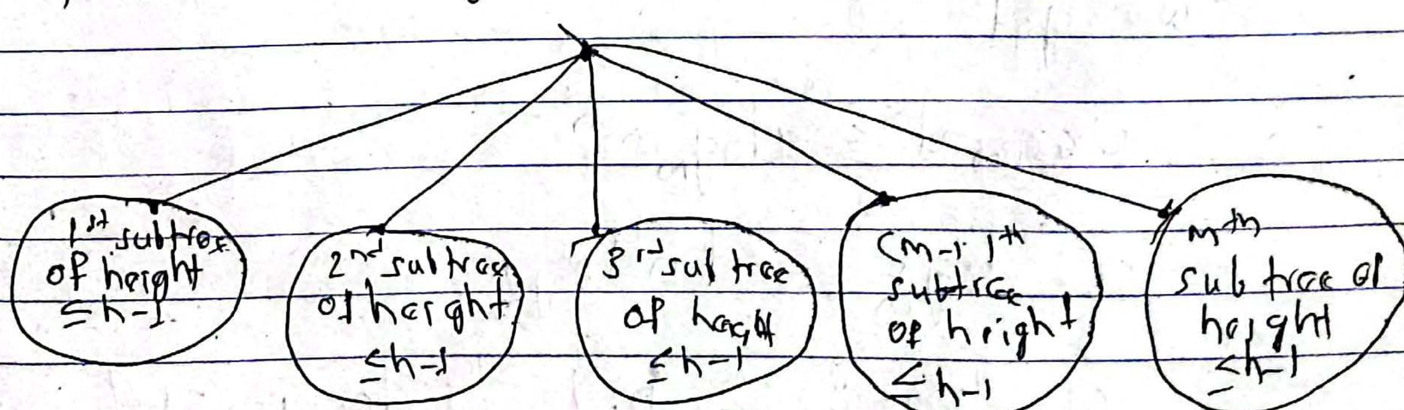
Proof :-

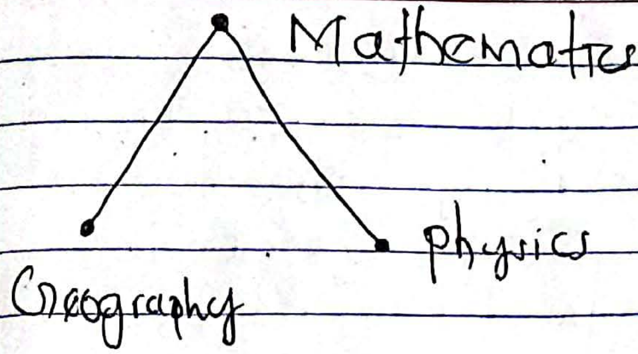
We prove this theorem by using mathematical induction on height h.

First, assume that m-ary of height 1. These trees consist of a root with no more than m-children, each of which is a leaf. Hence, there are no more than $m^1 = m$ leaves in m-ary tree.

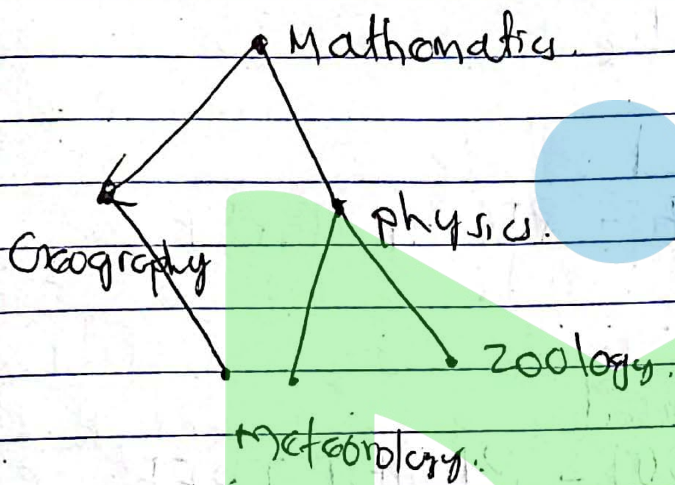
Now, assume that the result is true for all m-ary trees of height less than h.

Let T be an m-ary tree of height h. The leaves of T are the leaves of the subtrees of T obtained by deleting the edges from the root to each of the vertices at level 1, as shown figure.





place physics as right child of Mathematics. Since Geography comes before physics and mathematics then place Geography as the left child of mathematics. Since zoology comes after physics so place zoology as right child of physics.



Date
Page

*// Tree Traversal :

→ root

→ left sub tree

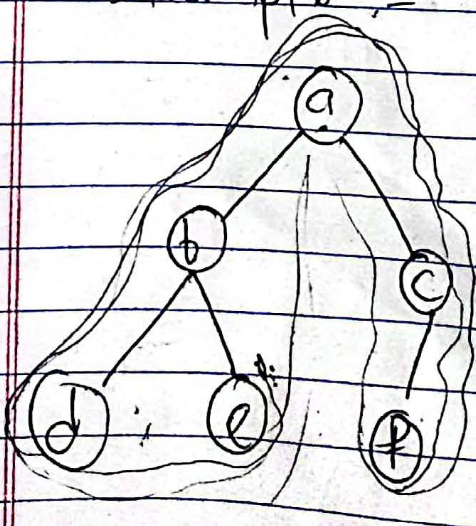
→ Right subtree

*// Pre order traversal
ROOT - LEFT TREE - RIGHT TREE

*// In order traversal
LEFT TREE - ROOT - RIGHT TREE

*// Post order traversal
LEFT TREE - RIGHT TREE - ROOT

*// Example :-



Date
Page

Pre-order traversal

We traverse tree T in preorder by first listing root a, followed by the preorder list of subtree with root b. Then we move to left subtree of b i.e. d there is no sub-tree of d. Then we move to right subtree of b i.e. e there is no subtree of e. Now we move on right subtree of a that contains c as the root and f as left sub tree.

Hence, pre order traversal is a b d e c f,

→ In order traversal

d b e a f c

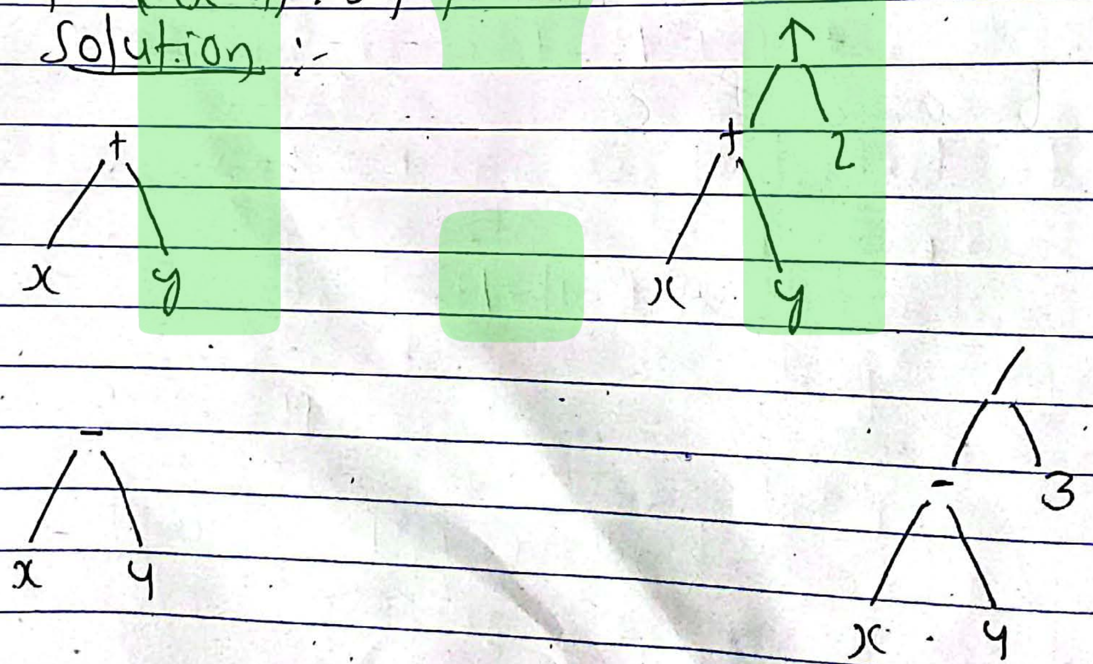
Date ______
Page ______

* Infix, Prefix and Post fix notation:-

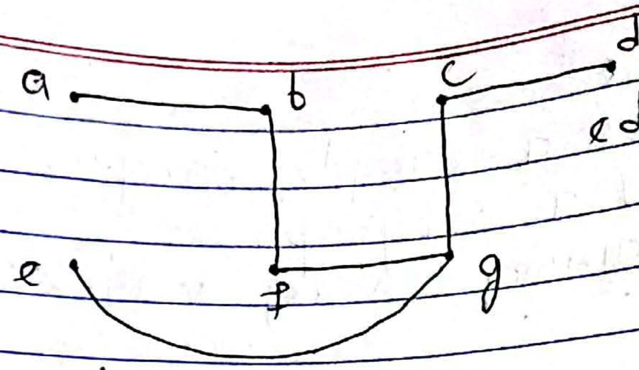
We can represent complicated expressions such as compound propositions, combination of sets and automatiz expressions using ordered rooted trees. For instance, consider the representation of an arithmetic expression involving the operations +, -, *, / and ↑ (exponent). We can use parenthesis to indicate order of the operation.

* Example:- What is the rooted tree that represents expression. $((x+y)\uparrow 2) + ((x-y)/3)$?

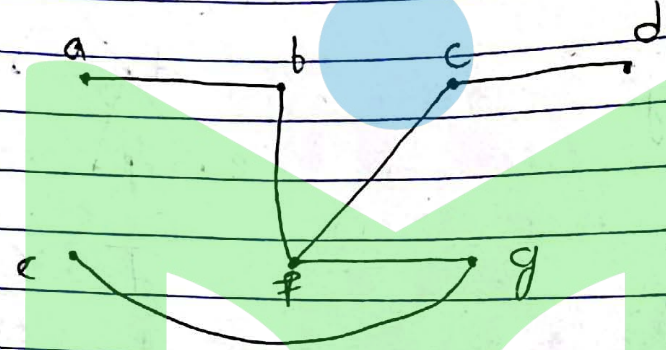
⇒ Solution:-



Date ______
Page ______



edge {e, f} is removed



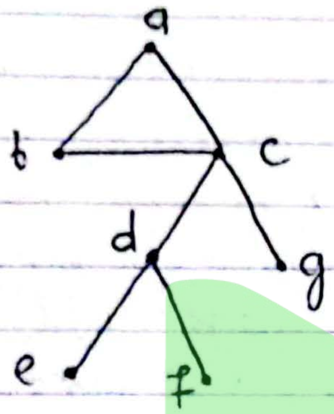
edge {c, g} is removed.

T

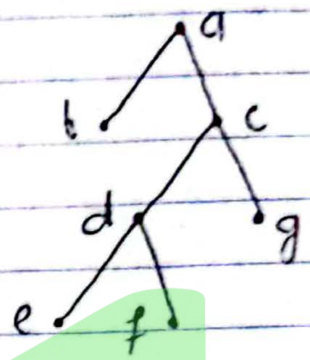
The graph G is connected, but it is not a tree because it contains simple circuits. Remove the edge {a, e}. This eliminates one simple circuit and resulting subgraph is still connected and have simple circuit. Next remove the edge {e, f} to eliminate a second simple circuit. Finally, remove edge {c, g} to produce a simple graph with no simple circuit. This subgraph is a spanning tree, because it is a tree that contains all vertices of graph 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.

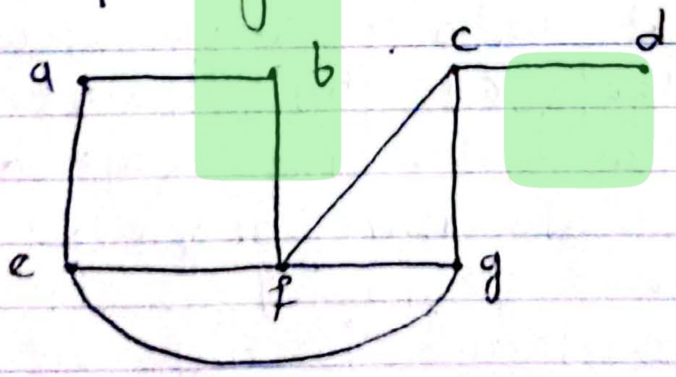


G

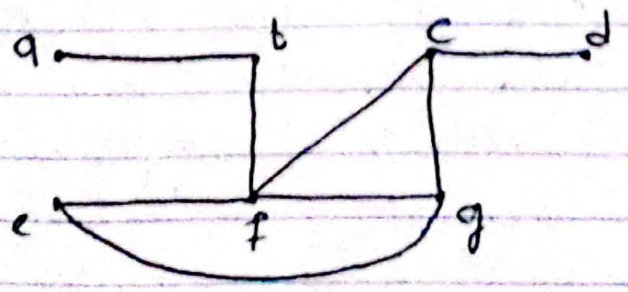


$T \subseteq G$

Example :- Find the spanning tree of Graph given below.



G



edge $\{a, e\}$ is removed

Date
Page

= − * 2 2 3

2 * 2 = 4

= − 4 3

4 − 3 = 1

∴ The value of the prefix expression − * 2 1 8 4 3 is 1.

* Example :- Find the value of post fix.

⇒ Solution :-

5 2 1 − − 3 1 4 + + *

2 − 1 = 1

= 5 1 − 3 1 4 + + *

5 − 1 = 4

= 4 3 1 4 + + *

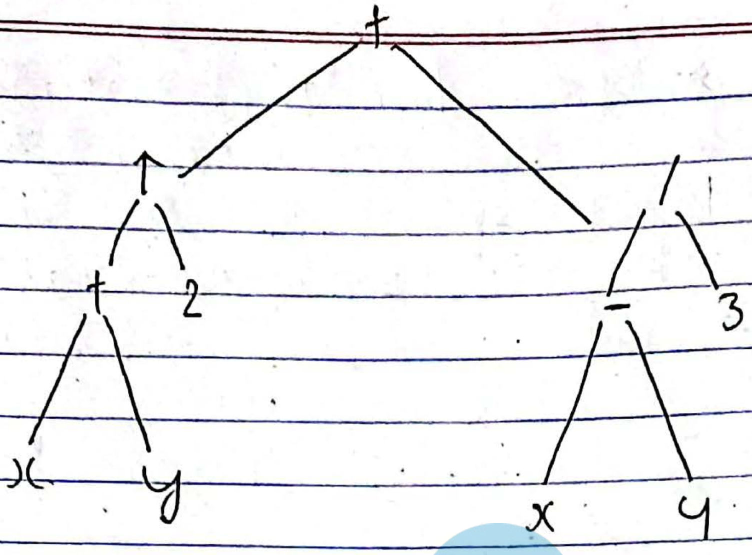
1 + 4 = 5

= 4 3 5 + *

3 + 5 = 8

= 4 8 *

4 * 8 = 32 //



The binary tree for this expression can be built from the bottom to up.

First, a sub-tree for the expression $x+y$ is constructed. Then, this is incorporated as the part of the longer subtree representing $(x+y)\uparrow 2$. Also, a subtree for $x-y$ is constructed and this is incorporated into a subtree representing $(x-y)/3$. Finally, subtrees representing $(x+y)\uparrow 2$ and $(x-y)/3$ are combined to form the ordered rooted tree representing $((x+y)\uparrow 2) + ((x-y)/3)$.

*/ Example:- Find the value of prefix expression $- * 2 / 8\ 4\ 3$.

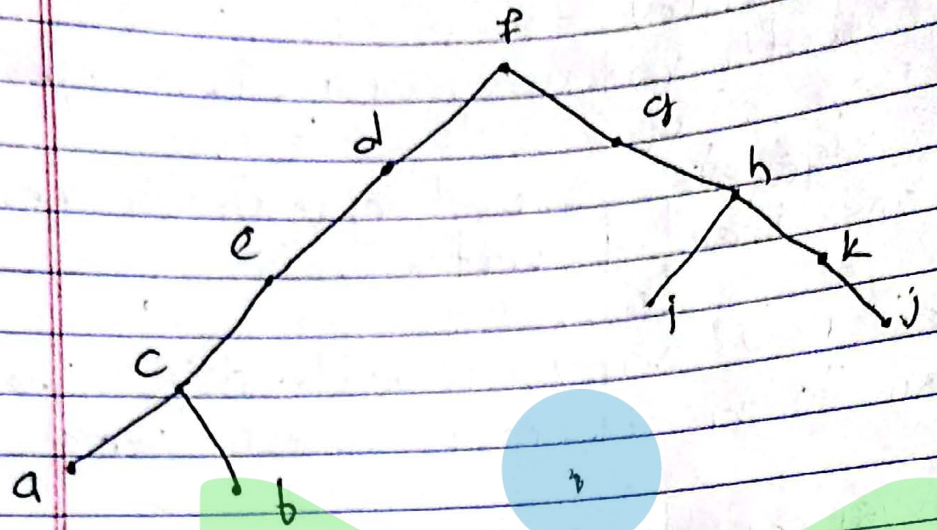
⇒ Solution :-

Given,

$- * 2 / 8\ 4\ 3$

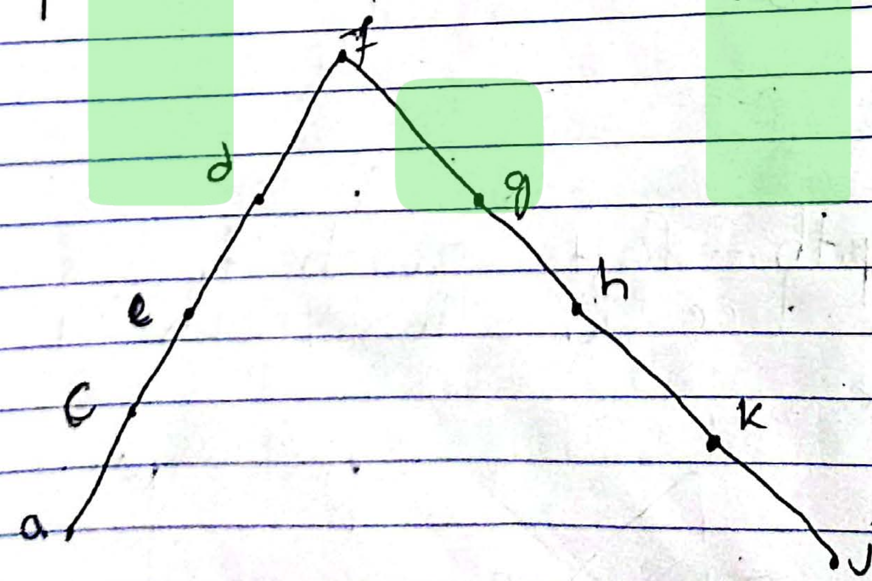
$8/4 = 2$

Date ____
Page ____



⇒ Solution :-

We arbitrary begin with vertex f. A path is built successively adding edges incident with vertices not already in the path, as long as this is possible. This produces a path f, g, h, k, j



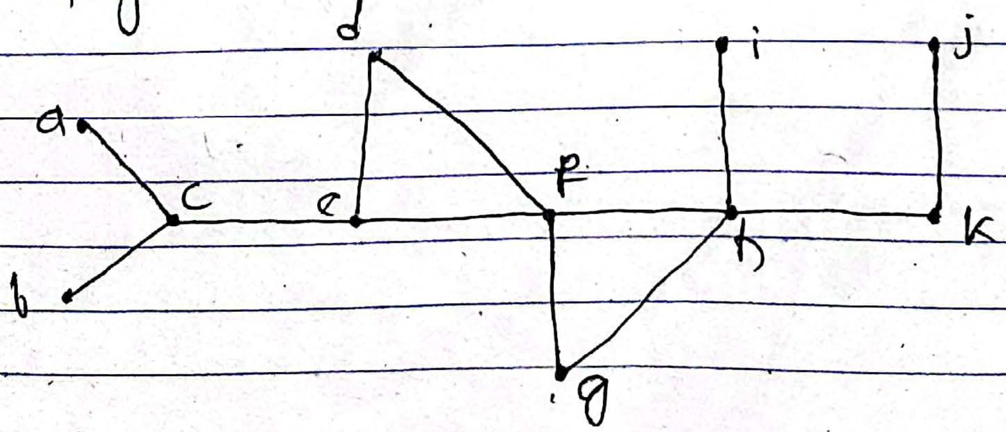
Date ______
Page ______

Explanation (Describe) :-

1. Choose a "root" vertex for a rooted graph.
2. Create an edge by connecting the current vertex to an incident vertex.
3. From the new vertex, repeat step 2, until you can no longer connect each vertex.
4. If all vertices have been visited, you have a spanning tree. If not go to the next to last visited vertex to form a new path from the vertex.
5. Continue visiting the next to last vertex of the current vertex until all vertices have been visited.

Example :-

Use Depth - first search to find a spanning tree for the graph G shown in ~~graph~~ Figure



every vertices of G. There is a path in T betn. any two of it's vertices. Because T is a subgraph of G, there is a path betn. any two of its vertices. Hence, G is connected.

Thus proved

(i) Depth-first search.
(ii) Breadth-first search

Algorithm 1 :- Depth-first search:-
PROCEDURE DFS (G: connected graph with vertices $v_1, v_2, \ldots, v_n$)
T := tree consisting only of the vertex v_1.
visit (v_1)
PROCEDURE visit (v: vertex of G)
For each vertex w adjacent to v and yet in T.
add vertex w and edge {v,w} to T
visit (w).

Date
Page

Theorem :- A simple graph is connected if and only if it has a spanning tree.

⇒ Proof :-

First suppose that a simple graph G is connected then we have to show that it has a spanning tree. If G is not a tree, it must contain a simple circuit. Remove an edge from one of these simple circuits. The resulting subgraph has one fewer number edge but still contain all the vertices of G. This subgraph is still connected because when two vertices are conn-ected by a path containing the removal edge they are connected by a path not containing this edge.

If this subgraph is not a tree, it has a simple circuit, so as before, remove an edge that is in a simple circuit. Repeat this proces until no circuit remains. This is possible because there are only a finite number of edges. This process terminates when no simple circuit remains. A tree is produced because the graph stays connected. This tree is a spanning tree because it contains all the vertices of G.

⇒ Conversely, assume that a simple graph G has a spanning tree 'T' then we have to prove that it is connected. 'T' contains

Date
Page

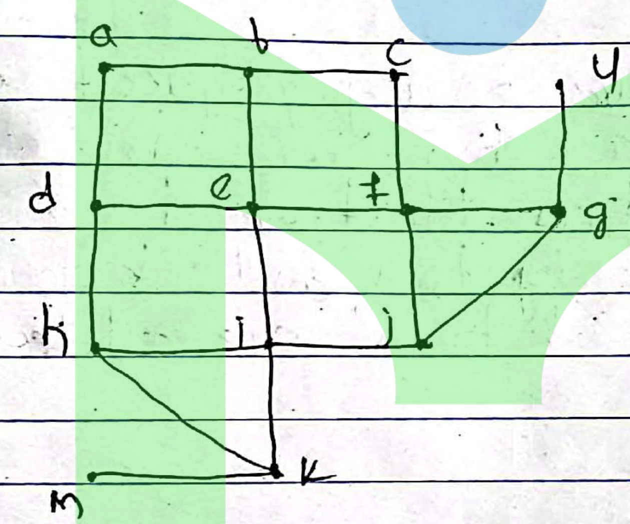
We choose the vertex e to be the root. Then we add edges incident with all vertices adjacent to e. So edge e to b, e to d, e to f and e to i we added. These four vertices are at level 1. Now, we begin with vertex b, the edges from b to a and c are added. Similarly, edges d to h, f to g and i to k are added. The vertices a, c, h, g, j and k are at level 2.

Since, there are no any adjacent vertices of vertices a, c, h, j. So these vertices are end vertices. Next, we have adjacent vertex from g to l and k to m, so edges g to l and k to m are added.

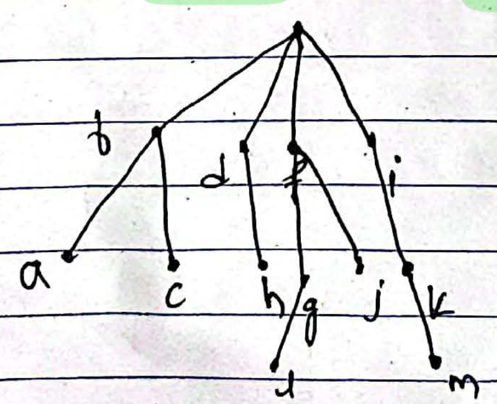
in the spanning tree.

4. Repeat adding levels and edges until all vertices have been visited.

* Example:- Use - Breadth - First search to find a spanning tree for the graph shown in given figure.



⇒ Solution:-



Date
Page

* Algorithm 2: Breadth-First Search:-

PROCEDURE BFS(G: connected graph with vertices $v_1, v_2, \ldots, v_n$)

T := tree consisting only a vertex v_1

L := empty list

Put v_1 in the list L of unprocessed vertices

WHILE L is not empty
remove the 'first' vertex v from L
FOR each neighbour w of v
IF w is not in L and not in T THEN
add w to the end of the last L and
w and {v, w} to T.

* Describe:-

1. Choose a root vertex for a rooted graph

2. Add all edges incident to the vertex. Then newly-connected vertices are level 1 in the spanning tree.

3. From each vertex in level 1 in order, add all edges incident to the vertices that are not already ~~inclined~~ included

Date
Page

Next, backtrack to k. There is no any path begining at k. Again backtrack to h there is a path from h to i. Next, backtrack to h and then backtrack to f via g. Now, there is a path from f to a i.e. f, d, e, c, a.

Next backtrack to c and there is a path from c to b. Now all the vertices of the given graph are visited and no any new path. Hence the obtained graph is a spanning tree of given graph.