Traversal of Tree

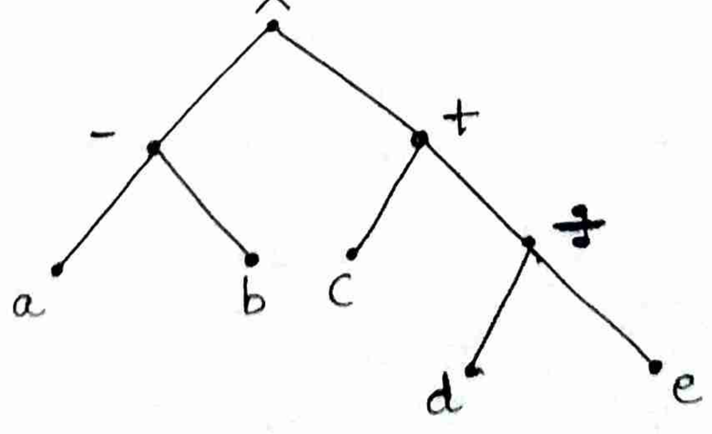
Traversing means to visit each vertex of tree exactly once. There are three standard ways of traversing a binary tree T with Root R. These algorithms are called preorder, inorder and post order traversals and are as follows:

I) Preorder :
1) Process the root r
2) Traverse the left subtree of r in preorder
3) Traverse the right subtree of r in preorder.

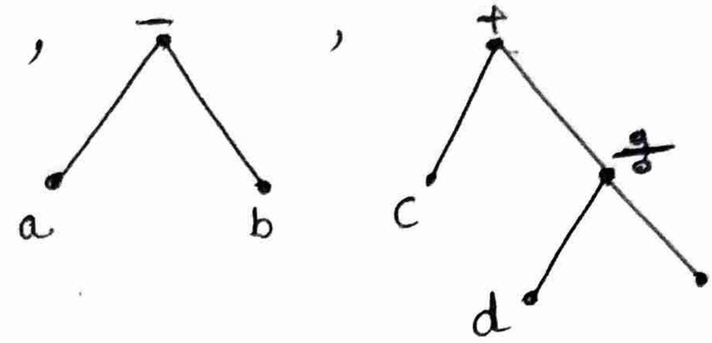
II) Inorder :
1) Traverse the left subtree of in inorder
2) Process the root r
3) Traverse the right subtree of r in inorder

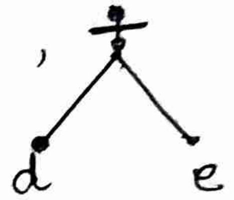
III) Post order :
1) Traverse the left subtree of r in postorder
2) Traverse the right subtree of r in postorder
3) Process the root r.

Ques:– Search the following Tree in preorder postorder and in-order.



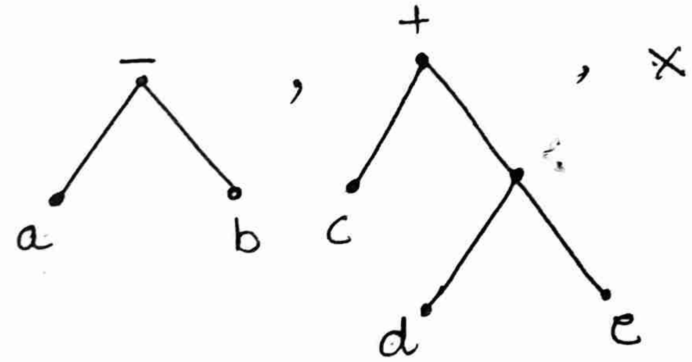
Pre Order

I) × , 

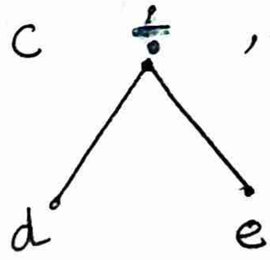
II) × , − , a , b , + , c , 

III) × , − , a , b , + , c , ÷ , d , e

Post Order

I) 

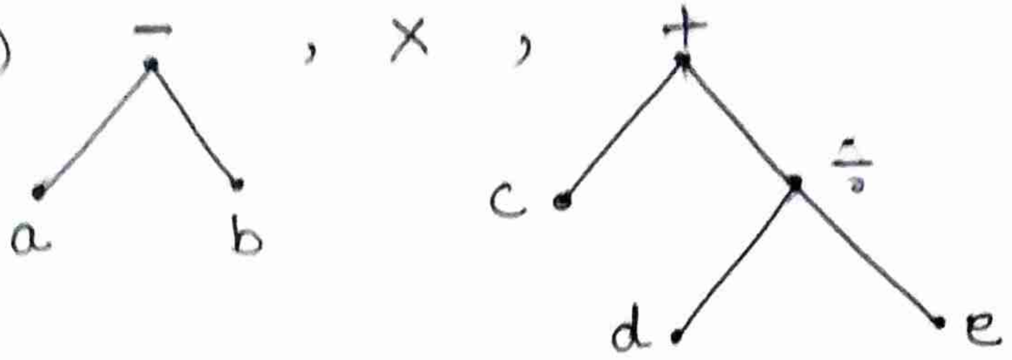
, ×

II) a, b, − , c 

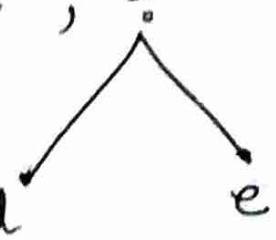
, + , ×

III) a, b, − , c , d , e , ÷ , + , ×

Inorder

I) 

II) a, −, b, ×, c, +, ÷



III) a, −, b, ×, c, +, d, ÷, e.

Prefix, Infix and Postfix

Prefix form: when a preorder traversal is performed on an expression tree then result obtained is called prefix form of the given algebraic expression.

Postfix Form: When a post-order traversal is performed on an expression tree then result obtained is called postfix form of the given algebraic expression

Infix Form: Infix form results from the in-order traversal of algebraic expression tree.

Q Evaluate prefix expression.

$+ - * 2\ 3\ 5 / \uparrow 2\ 3\ 4$

soln

I) $+ - \underbrace{* 2\ 3}_{2*3=6}\ 5 / \uparrow 2\ 3\ 4$

II) $+ \underbrace{- 6\ 5}_{6-5=1} / \uparrow 2\ 3\ 4$

III) $+ 1 / \underbrace{\uparrow 2\ 3}_{2\uparrow 3=8}\ 4$

IV $+ 1 \underbrace{/ 8\ 4}_{8/4=2}$

V) $\underbrace{+ 1\ 2}_{1+2=3}$

VI) 3

Q: What is the value of post fix expression

$7\ 2\ 3 * - 4 \uparrow 9\ 3 / +$

sol

I) $7 \underbrace{2\ 3 *}_{2\times 3=6} - 4 \uparrow 9\ 3 / +$

II) $\underbrace{7\ 6 -}_{7-6=1} 4 \uparrow 9\ 3 / +$

III) $\underbrace{1\ 4 \uparrow}_{1\uparrow 4=1} 9\ 3 / +$

IV) $1 \underbrace{9\ 3 /}_{9/3=3} +$

V) $\underbrace{1\ 3 +}_{1+3=4}$

VI 4

Spanning Tree

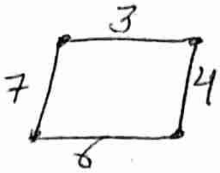
Let G be connected graph. A subgraph T of G is called spanning tree if
(i) T is tree
(ii) T contains all vertices of G

For e.g.  is a connected graph G. clearly it is not a tree.

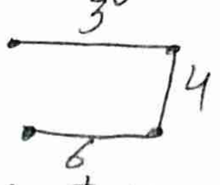
A spanning tree of G is 

* Spanning tree is not unique.

Minimal Spanning Tree:- A minimal spanning tree of a weighted graph is a spanning tree with the condition that sum of tree is as small as possible.

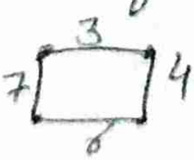
E.g Given weighted Graph is 

Minimal spanning tree of this graph is



weight of minimal spanning tree = 3+4+6 = 13.

Maximal Spanning Tree:- A maximal spanning tree of a weighted graph is a spanning tree with the condition that the sum of weights of tree is as large as possible.

E.g Given weighted Graph is 

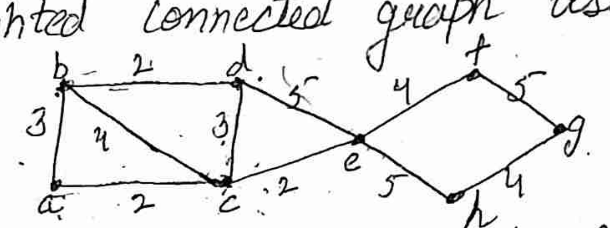
Maximal Spanning tree is , weight = 7+6+4=17

Kruskal's Algorithm to find Minimal Spanning Tree

Let G be the given connected graph with n-vertices. Then Kruskal's Algorithm to find minimal spanning Tree involves the following steps:-

1. Write all the edges of graph in increasing order of their weight.
2. Select the smallest edge of G.
3. For each successive step select another smallest edge of G which makes no cycle with previously selected edges.
4. Go on repeating step 3 until n-1 edges have been selected. The sum of weights of these n-1 edges will constitute required minimal spanning tree.

Ques? Find the minimal spanning tree for the following weighted connected graph using Kruskal's Algorithm



Sol^n First we write all the edges in Increasing order of weight i.e $E = \{ac, bd, ce, ab, cd, bc, ef, gh, ed, eh, fg\}$.

No. of vertices (n) = 8

we start from edge ac & then select edges one by one from E until we select 7 edges i.e (n-1) edges

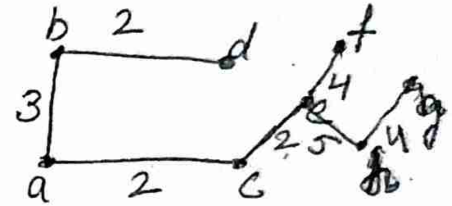
OR

Soln 17) No. of vertices $(n) = 8$

No. of edges include $(n-1)$ ie 7.

write down the edges in - increasing order, we get

Edges	weight	Added / Not added
(a,c)	2	Added
(b,d)	2	Added
(c,e)	2	Added
(a,b)	3	Added
(c,d)	3	Not Added
(b,c)	4	Not Added
(e,f)	4	Added
(g,h)	4	Added
(e,d)	5	Not added
(e,h)	5	Added
(f,g)	5	Not Added



In The above graph, we selected 7 edges, so we stop algorithm. Minimum Spanning Tree is as shown above & sum of weights is $2+2+2+3+4+4+5 = 22$