Operations on Binary Tree - Part I

Course on C-Programming & Data Structures: GATE - 2024 & 2025

Vishvadeep Gothi • Lesson 43 • Jan 20, 2023

Data Structure
Tree 2

By: Vishvadeep Gothi

Tree Terminologies

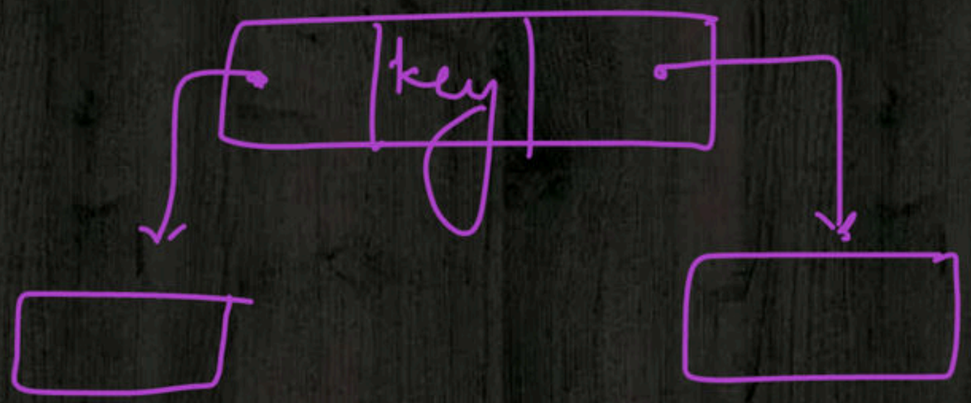
- Root
- Child
- Parent
- Sibling
- Subtree
- Internal Node
- External Node
- Level

Binary Tree

↓

every node => max 2 children

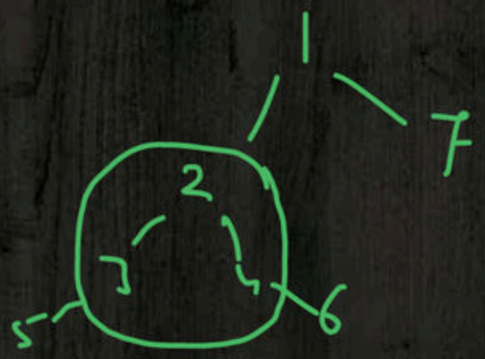
Linked representation

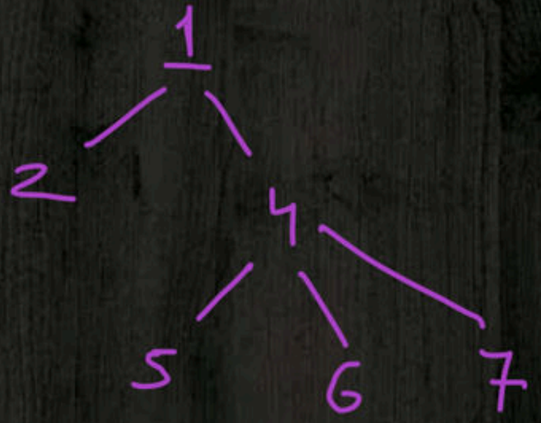


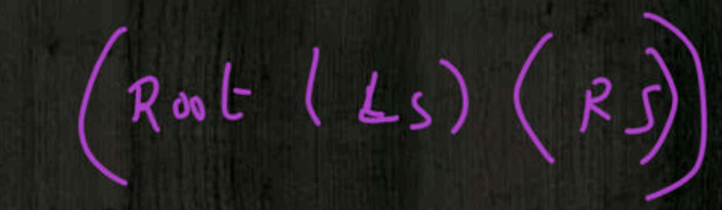
Question GATE-2000

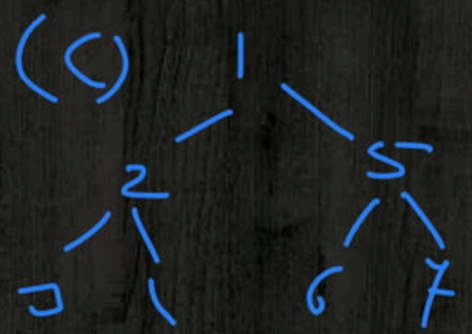
Consider the following nested representation of binary trees: (X Y Z) indicates Y and Z are the left and right subtrees respectively of node X. Note that Y and Z may be NULL or further nested. Which of the following represents a valid binary tree?

- (1 2 (4 5 6 7))
- (1 ((2 3 4) 5 6) 7)
- (1 (2 3 4)(5 6 7))
- (1 (2 3 NULL)(4 5))





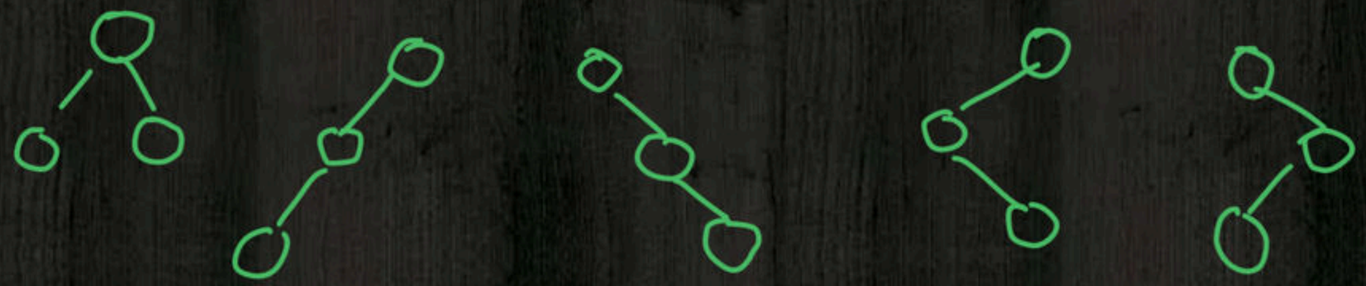




(3)

Question

How many binary tree can be constructed using 3 unlabeled nodes?



Ans = 5

---

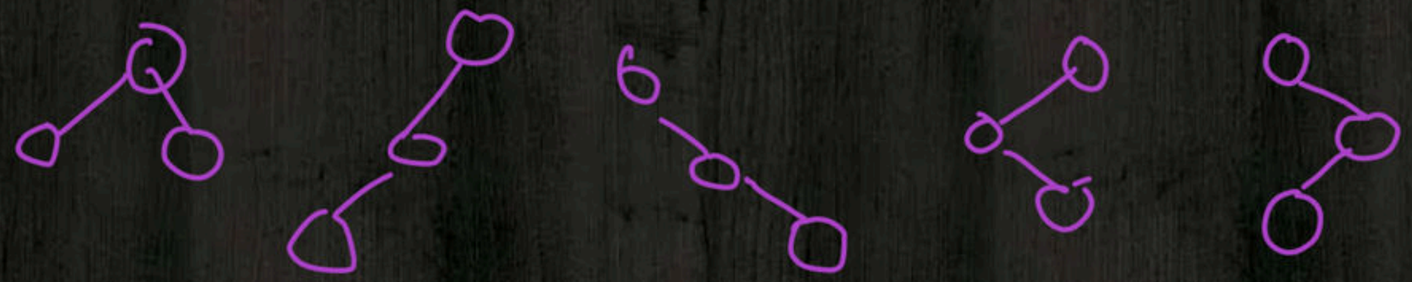
for n unlabeled nodes $\Rightarrow$

no of distinct BTs $= \frac{^{2n}C_n}{n+1}$

Question

Distinct

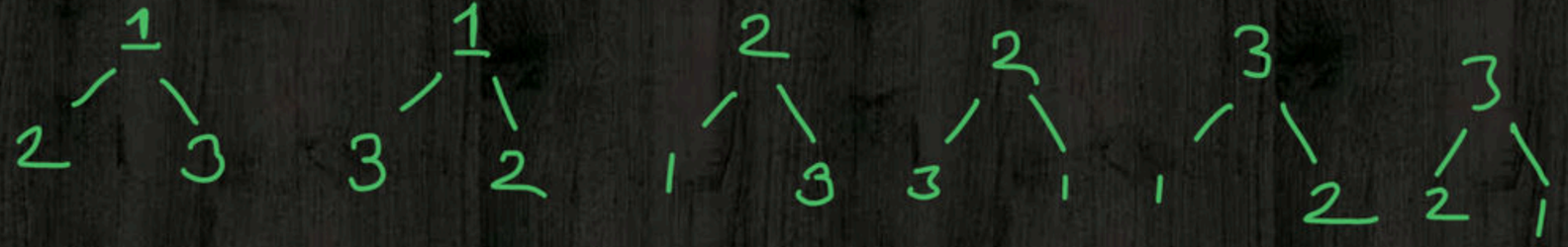
How many binary tree can be constructed using 3 distinct keys?



Ans = 5 * 6

= 30

keys ⇒ 1, 2, 3



n! ways to populate one BT with n keys

for n distinct keys,

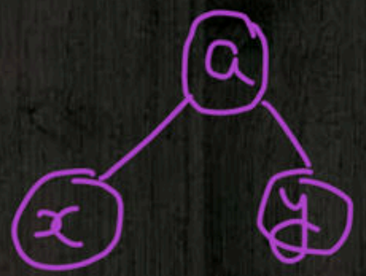
no of distinct BST $\Rightarrow \frac{{}^{2n}C_n}{n+1} * n!$

Tree Traversals

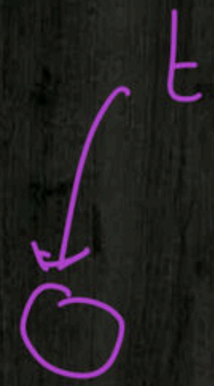
① Preorder Traversal ⇒ nLR ⇒ axy

② Inorder —11— ⇒ LnR ⇒ xay

③ Postorder —11— ⇒ LRn ⇒ xya

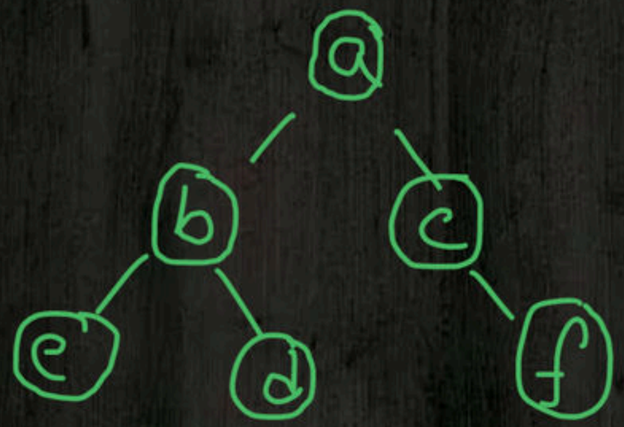


Preorder




```
void Preorder(struct BTNode *t)
{
    if(t)
    {
        printf("%d", t→data);
        Preorder(t→LeftChild);
        Preorder(t→RightChild);
    }
}
```

Question

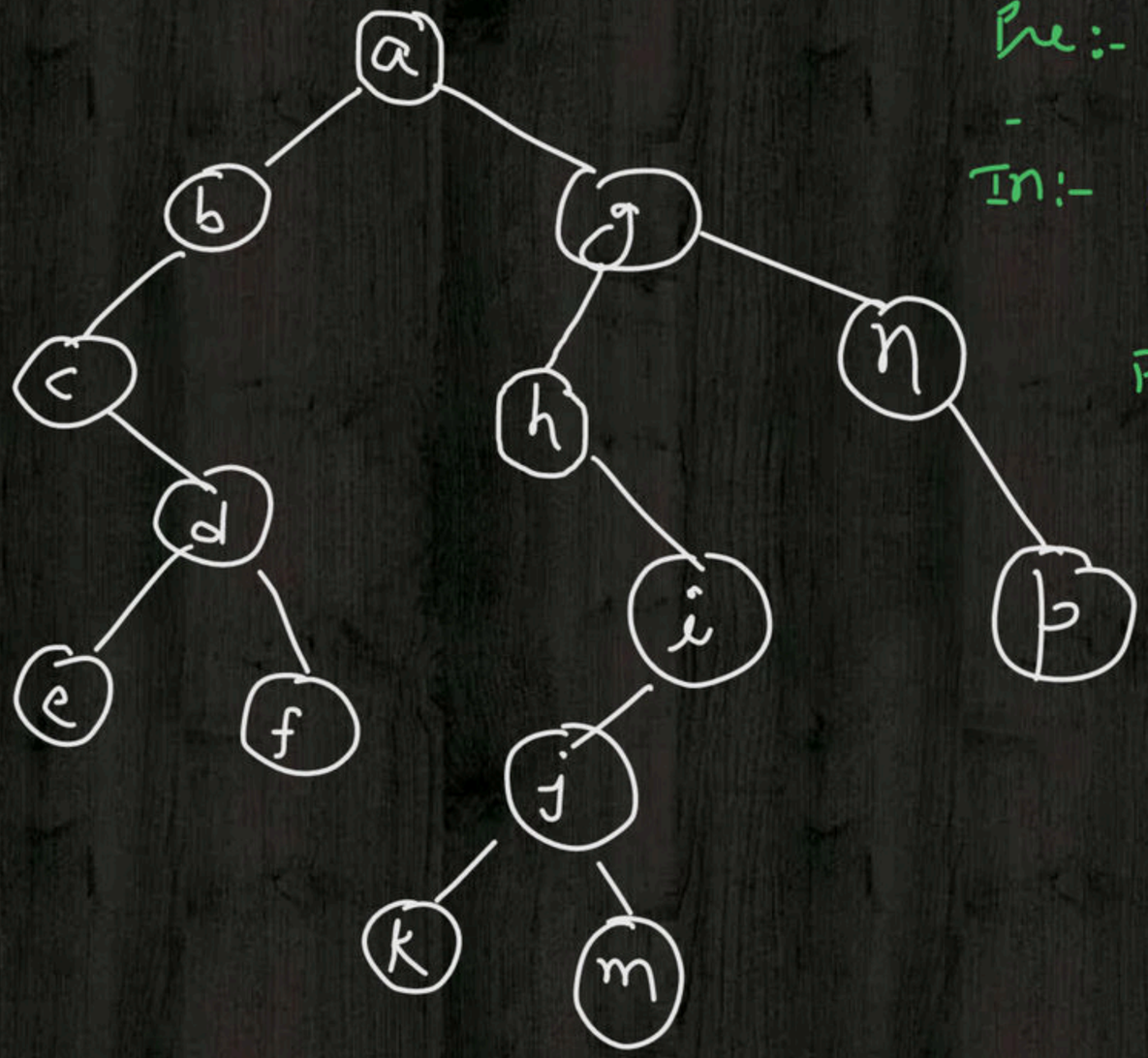


Pre :- a b e d c f

In :- e b d a c f

Post :- e d b f c a

Question



Pre :- a b c d e f g h i j k m n p

In :- c e d f b a h k j m i g n p

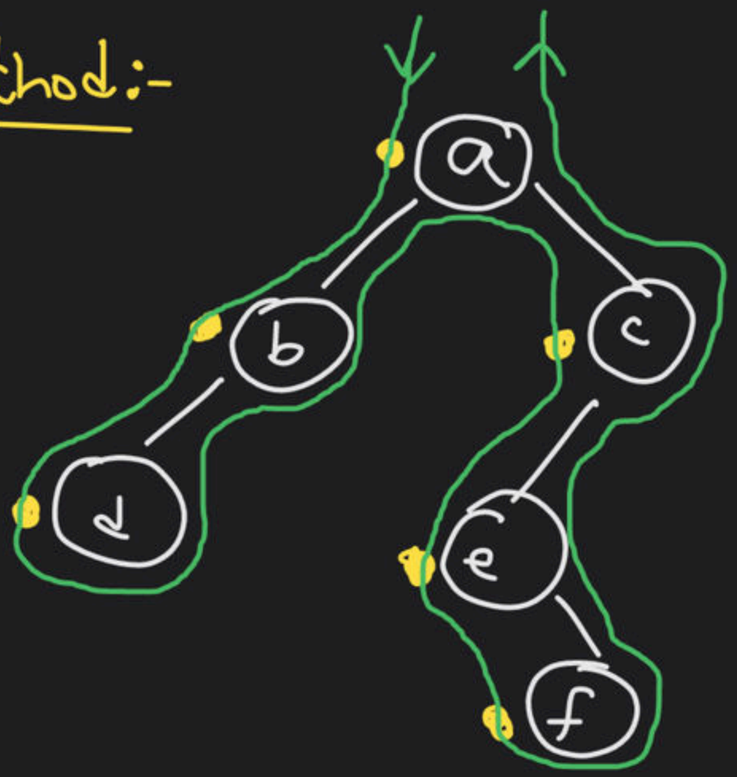
Post :- e f d c b k m j i h p n g a

Preorder:-

node traversed before Left & right subtree

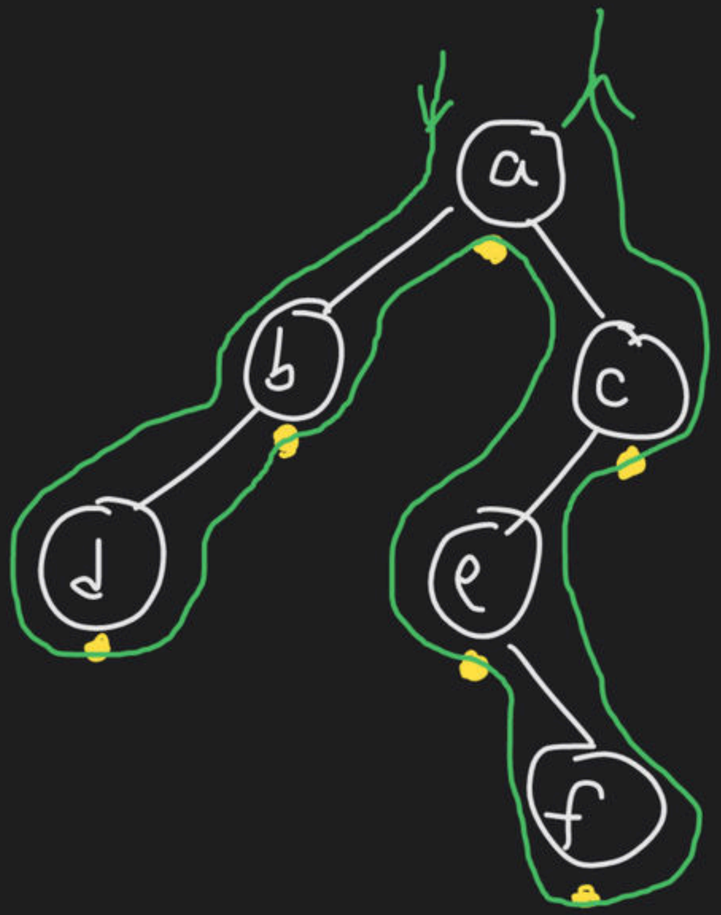
Traversal direction ⇒ Left to right

Dot method:-



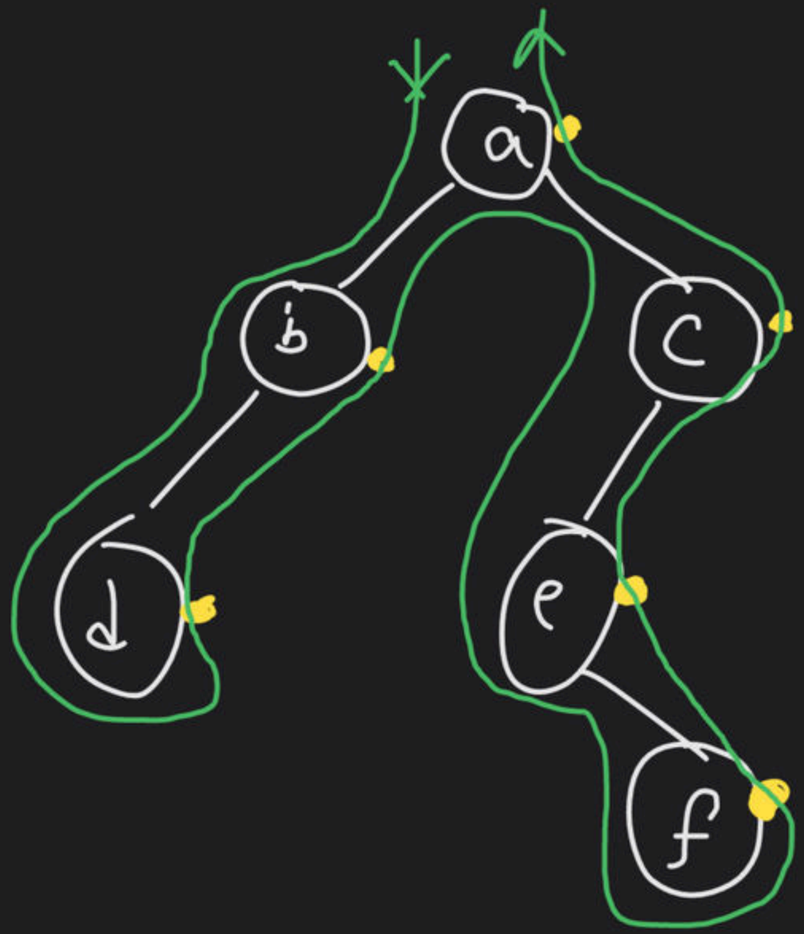
a b d c e f

Inorder:- node traversed in b/w left & right subtree



In:- d b a e f c

Postorder:- node is traversed after Left & Right subtree



d b f e c a

Observation from Traversal

⇒ First symbol of preorder traversal ⇒ Root of tree

⇒ Last —"— postorder —"— ⇒ Root of tree

Question

Identify the inorder, preorder and postorder traversal:

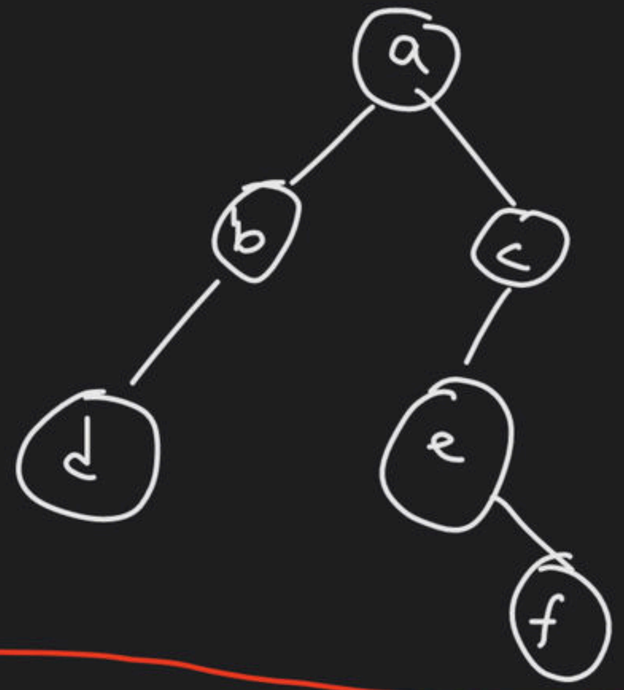
1. GHFEAPKDCBXM → Post
2. EGFHAPMCKDXB — In
3. MPAEFGHXCDKB — Pre

Converse Order Traversal

Converse Preorder ⇒ nRL

Converse Inorder ⇒ RnL

Converse postorder ⇒ RLn



Converse Preorder :- a c e f b d

converse Inorder :- c f e a b d

Converse Postorder :- f e c d b a

conventional

Preorder :- a b d c e f

Inorder :- d b a e f c

Postorder :- d b f e c a

```
void ConverseBreorder(struct BTnode *t)
{
   if (t)
   {
      printf("%c", t -> data);
      conversepreorder(t -> Rightchild);
      Conversepreorder(t -> Leftchild);
   }
}
```

Observation from Converse Order Traversal

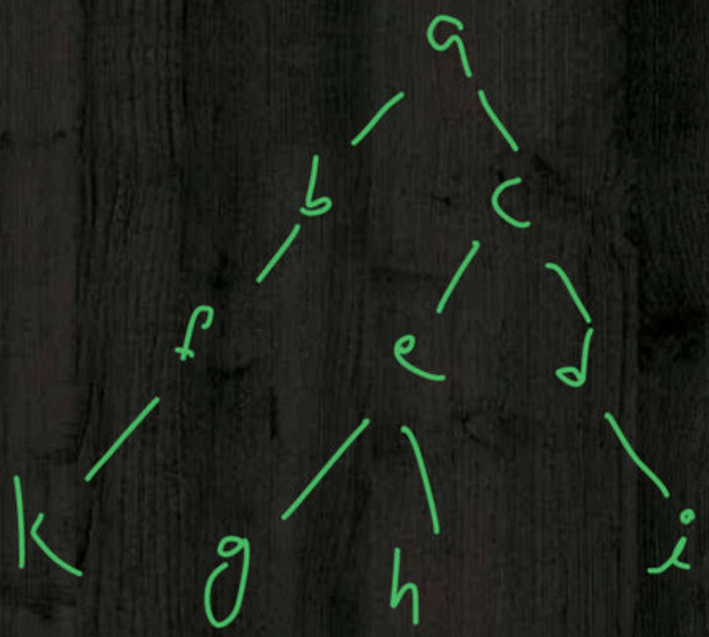
Reverse of converse preorder traversal ⇒ Conventional postorder

—— " —————— postorder —" —— ⇒ —" —— preorder

—— " —————— inorder —" —— ⇒ —" —— Inorder

Level Order Traversal

Starting from level 0, traverse all nodes of level no. L before traversing any node of higher level no.



a, b, c, f, e, d, k, g, h, i

a | b, c | f, e, d | k, g, h, i |

2! 3! 4!

no of distinct level order traversals possible = 1 * 2! * 3! * 4! = 288

Data Structure:
Tree 3

By: Vishvadeep Gothi

Hello!

I am Vishvadeep Gothi

I am here because I love to teach

Question

How many binary tree can be constructed using 3 unlabeled nodes?

Question

How many binary tree can be constructed using 3 distinct keys?

Conversion of General Tree to Binary

Tree Traversals

Preorder

```
void Preorder(struct BTNode *t)
{
    if(t)
    {
        printf("%d", t→data);
        Preorder(t→LeftChild);
        Preorder(t→RightChild);
    }
}
```

Question

Question

Observation from Traversal

Question

Identify the inorder, preorder and postorder traversal:

1. GHFEAPKDCBXM
2. EGFHAPMCKDXB
3. MPAEFGHXCDKB

Converse Order Traversal

Observation from Converse Order Traversal

Level Order Traversal

Constructing the Tree Using Traversals

To construct unique BT from traversals

- (1) Atleast 2 traversals are needed
- (2) one should be inorder

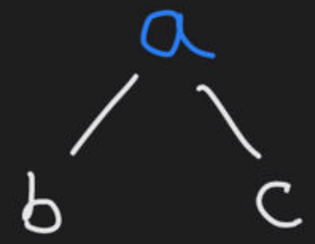
---

(1) Preorder or Postorder => Identify root of tree (subtree)

(2) Inorder => Identify left & right subtree

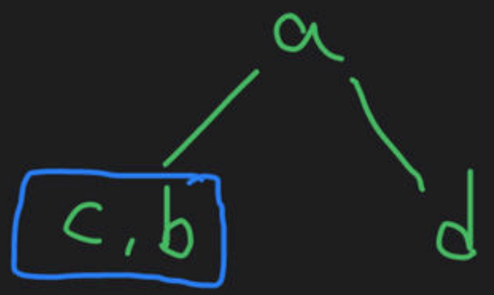
Pre:- a b c

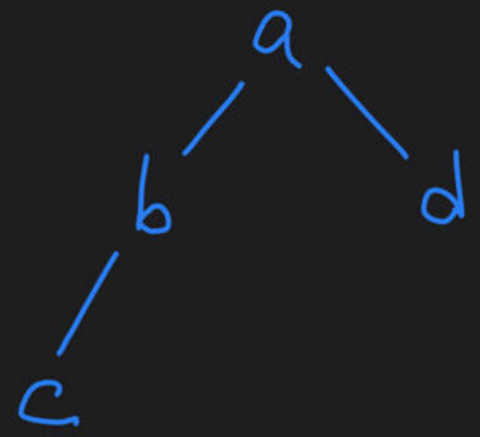
In:- b a c



Pre :- a b c d

In :- c b a d

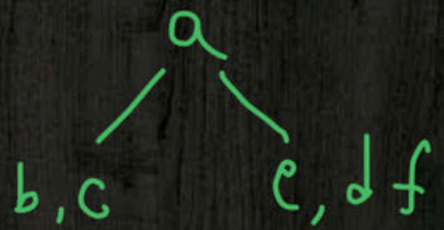


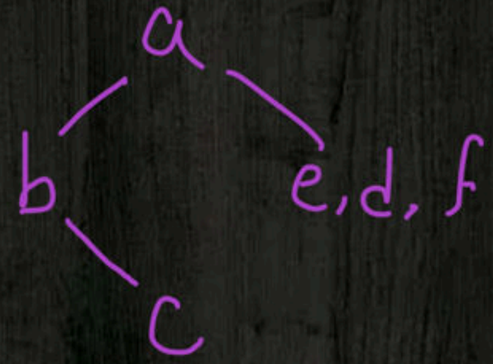


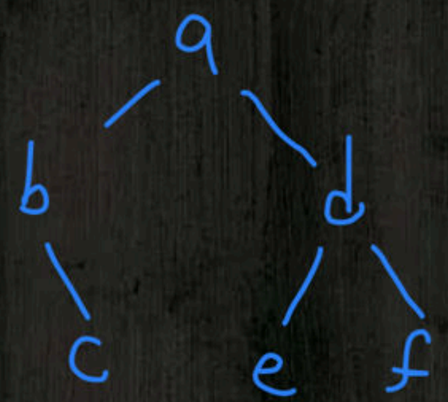
Question

PRE: abcdef

IN: bcaedf







Question

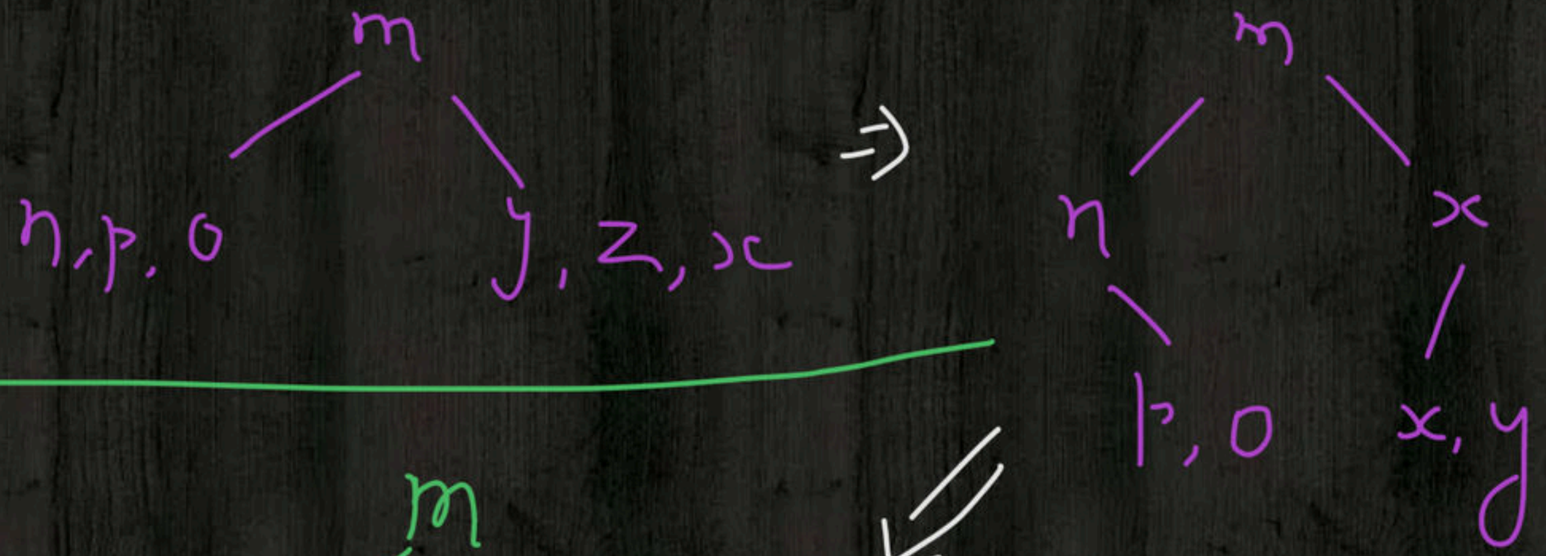
POST: efbgdca

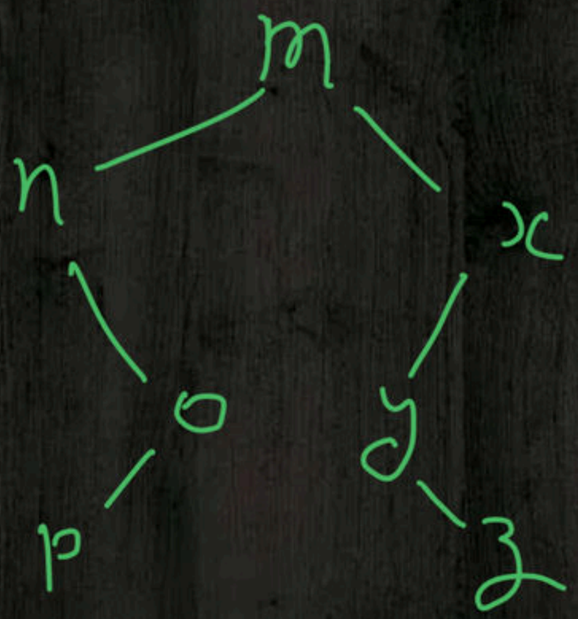
IN: ebfadgc

Question

PRE: mnopxyz

IN: npomyzx

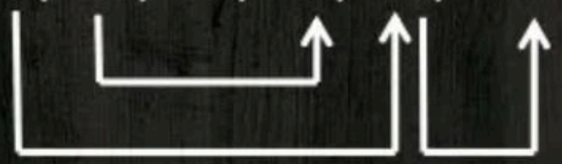




Question

Preorder: 1, 2, 4, 5, 3, 6

RightPointer:



Question

Postorder: 5, 6, 2, 7, 4, 3, 1

LeftPointer:

Question

Postorder: d e b f g c a
Degree: 0 0 2 0 0 2 2

Get the Tree and Inorder/Preorder traversal

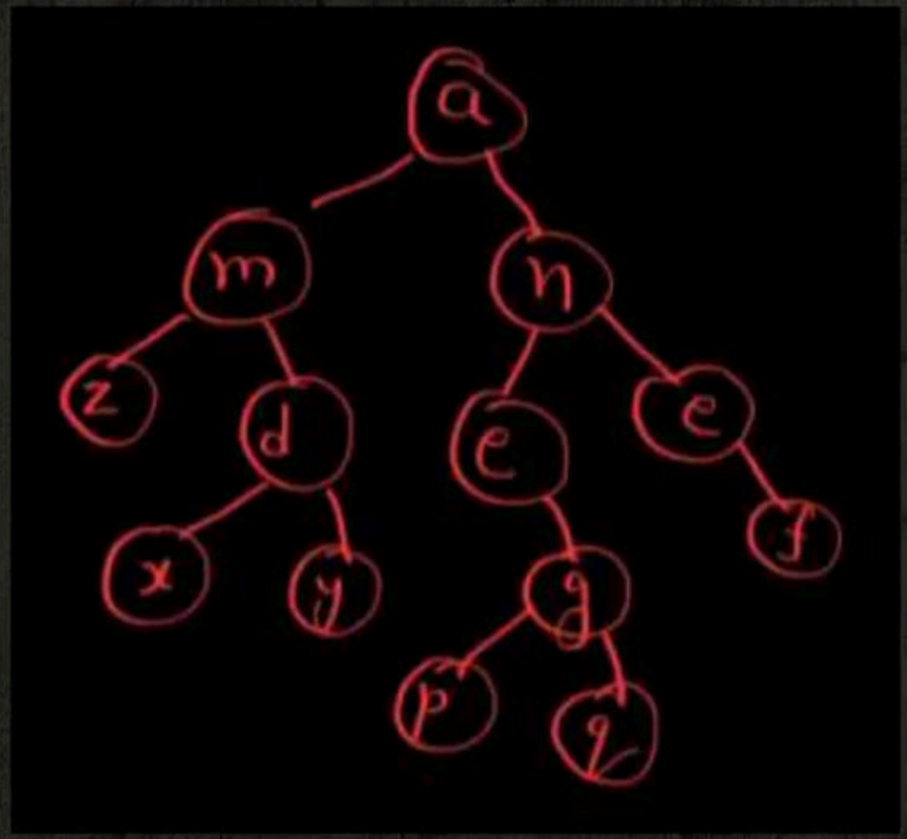
Conversion of General Tree to Binary

DPP

Question

Find Traversals

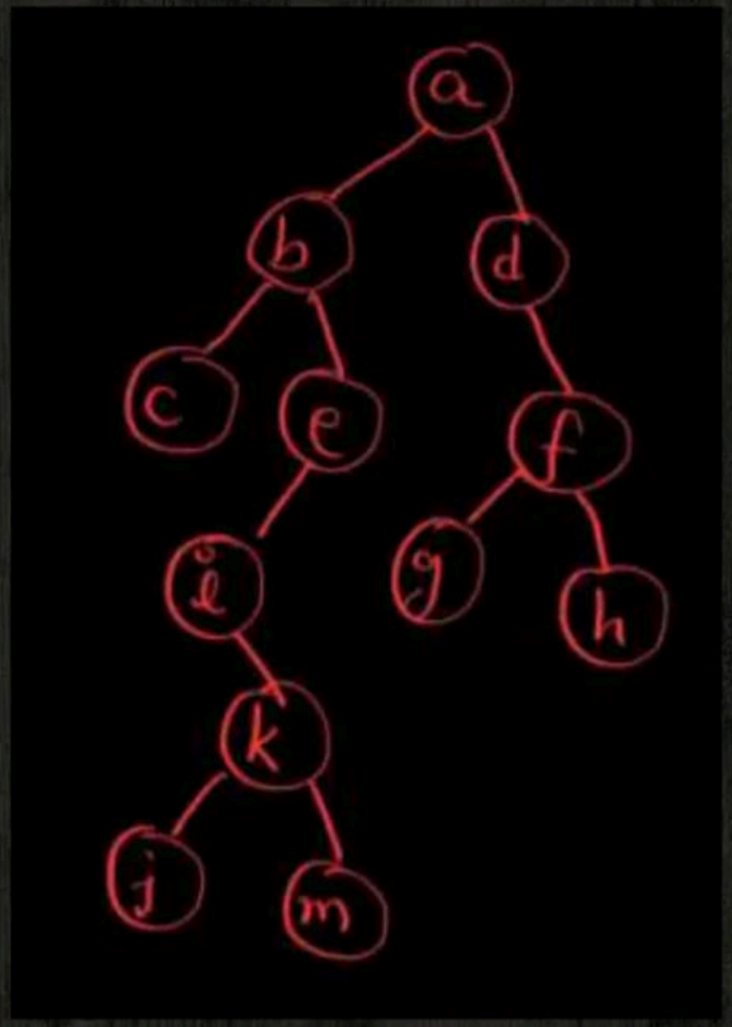
1. Preorder
2. Inorder
3. Postorder



Question

Find Traversals

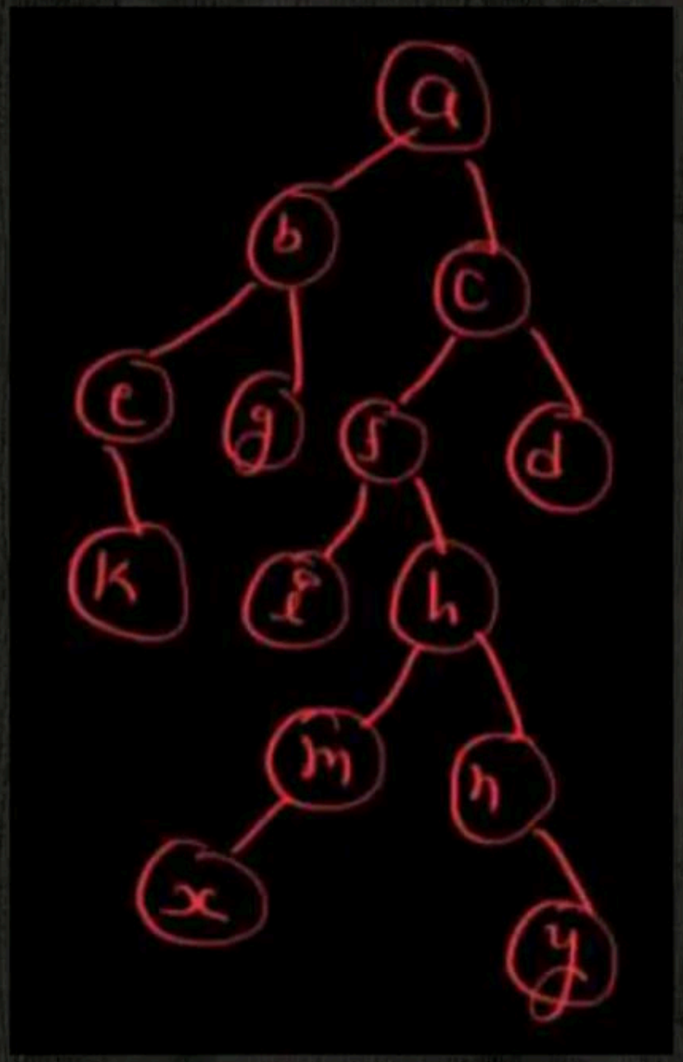
1. Preorder
2. Inorder
3. Postorder



Question

Find Traversals

1. Converse Preorder
2. Converse Inorder
3. Converse Postorder



Question

Draw the tree for:

Preorder: 1 2 3 4 5 6 7

Inorder: 2 4 3 1 6 7 5

Question

Postorder: dxcbygfea

Inorder: bdcxayfge

Question

Postorder: MBCAFHPYK

Inorder: MABCKYFPH

Question

Converse Postorder: 9, 10, 8, 7, 3, 6, 5, 4, 2, 1

Converse Inorder: 7, 9, 8, 10, 3, 1, 6, 4, 5, 2

Question

Converse Preorder: a, g, h, i, k, j, b, c, d, f, e

Inorder: c, e, d, f, b, a, h, j, i, k, g

Question

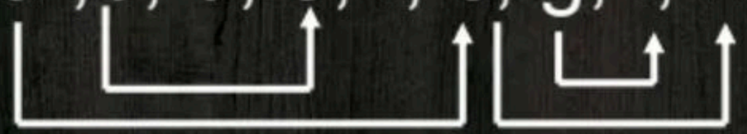
Preorder: 1, 2, 3, 4, 5, 12, 13, 6, 7, 8, 9, 11, 10

Converse Inorder: 10, 8, 11, 9, 7, 1, 2, 6, 4, 5, 13, 12, 3

Question

Preorder: a ,b, d, e, f, c, g, i, h

RightPointer:



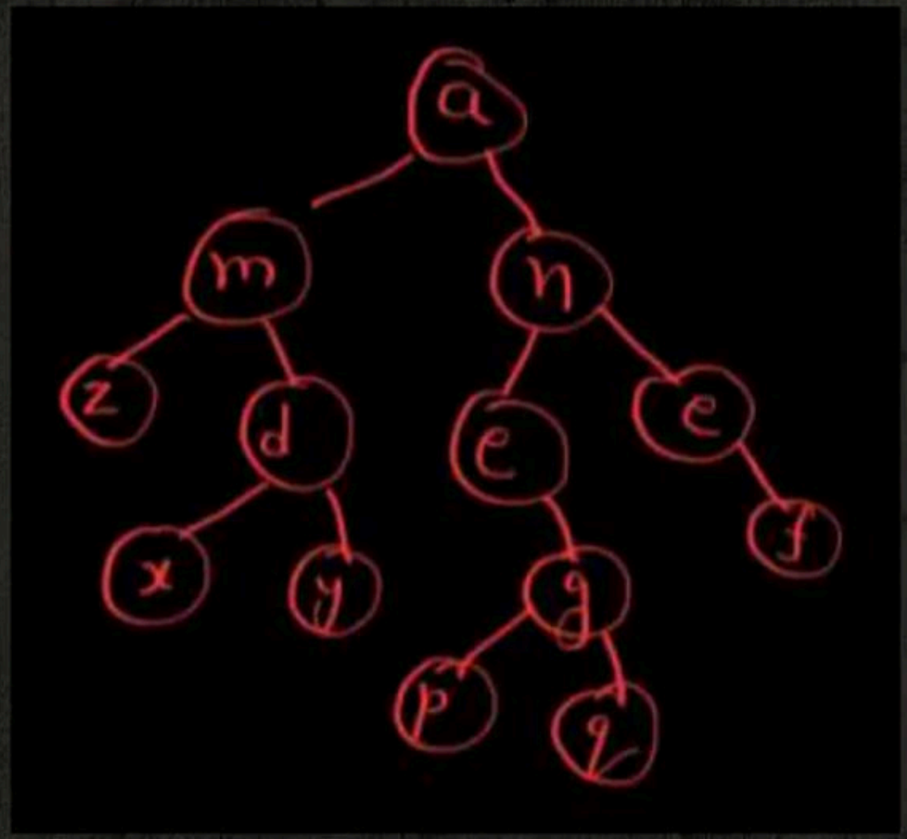
Happy Learning

DPP

Question

Find Traversals

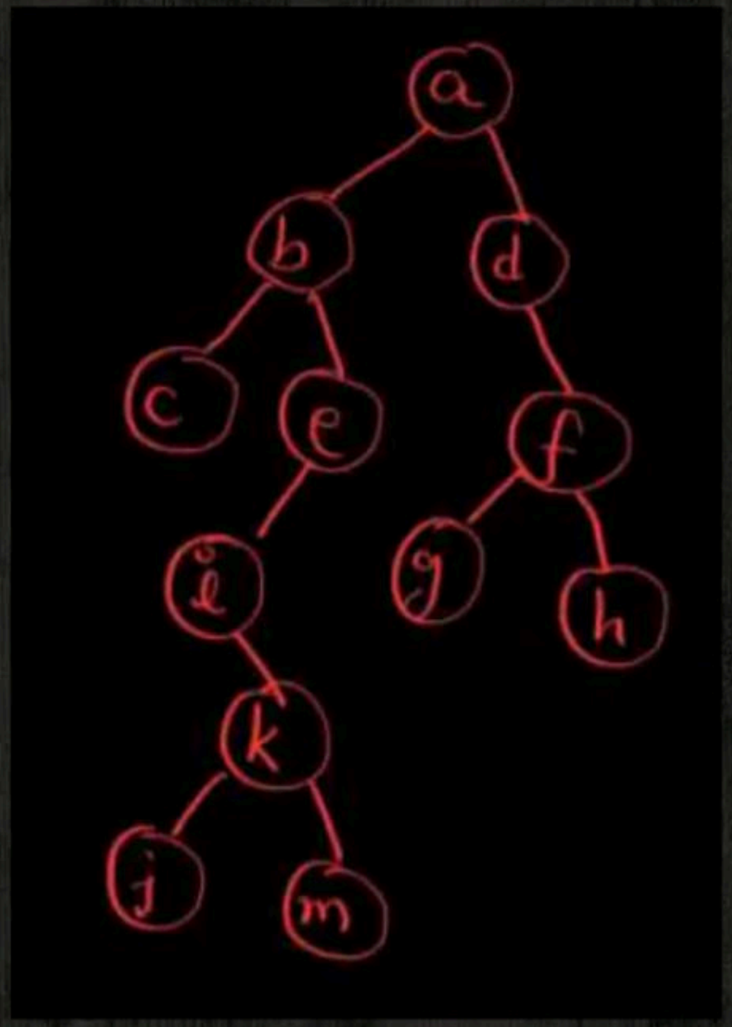
1. Preorder
2. Inorder
3. Postorder



Question

Find Traversals

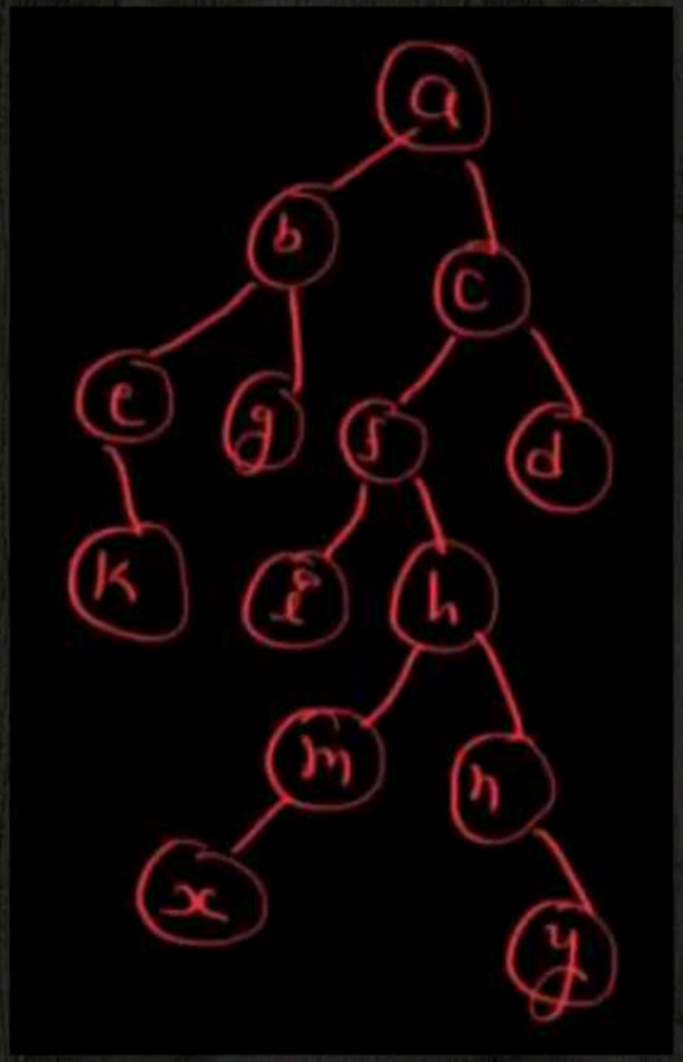
1. Preorder
2. Inorder
3. Postorder



Question

Find Traversals

1. Converse Preorder
2. Converse Inorder
3. Converse Postorder



Happy Learning