Binary Tree Construction

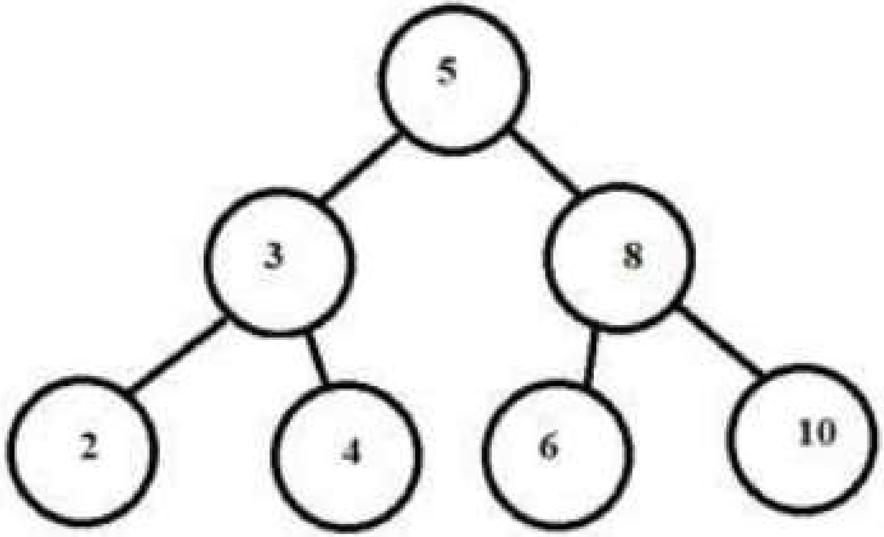
Construct a binary search tree from inorder and preorder traversal

Inorder Traversal: 2 3 4 5 6 8 10 (Left->Root->Right)

Pre-order Traversal: 5 3 2 4 8 6 10 (Root->Left->Right)

- As we know that preorder visits the root node first then the first value always represents the root of the tree. From above sequence 5 is the root of the tree.
- *Take the elements from Pre-order (starting from left side) and find its position w.r.t Root using In-order traversal.*

Contd…



The same approach is used for the construction of **binary tree**.

Complexity for the construction of BST = O(nlogn)

Complexity for the construction of Binary Tree = $O(n^2)$

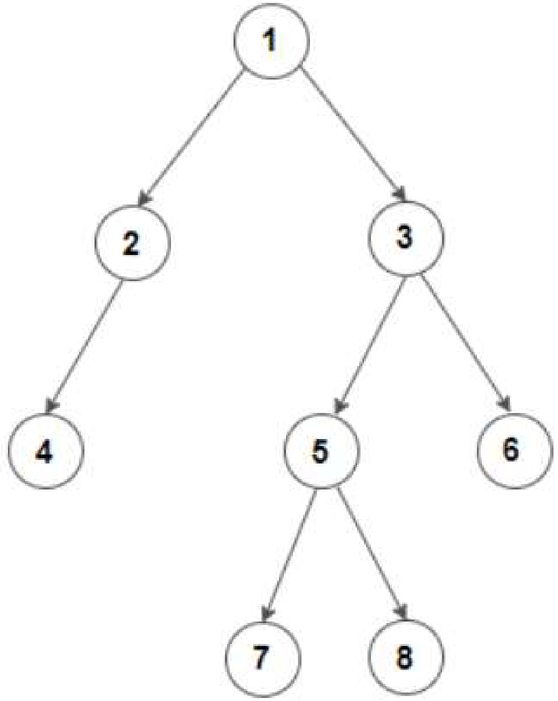
Construct Binary Tree from Inorder and Postorder Traversal

In-order Traversal: 4, 2, 1, 7, 5, 8, 3, 6 (Left-> Root-> Right)

Post-order Traversal: 4, 2, 7, 8, 5, 6, 3, 1 (Left->Right-> Root)

- ❖ As we know that post-order visits the root node at last then the last value always represents the root of the tree. From above sequence 1 is the root of the tree.
- ❖ *Take the elements from Post-order (starting from right side) and find its position w.r.t Root using In-order traversal.*

Contd…



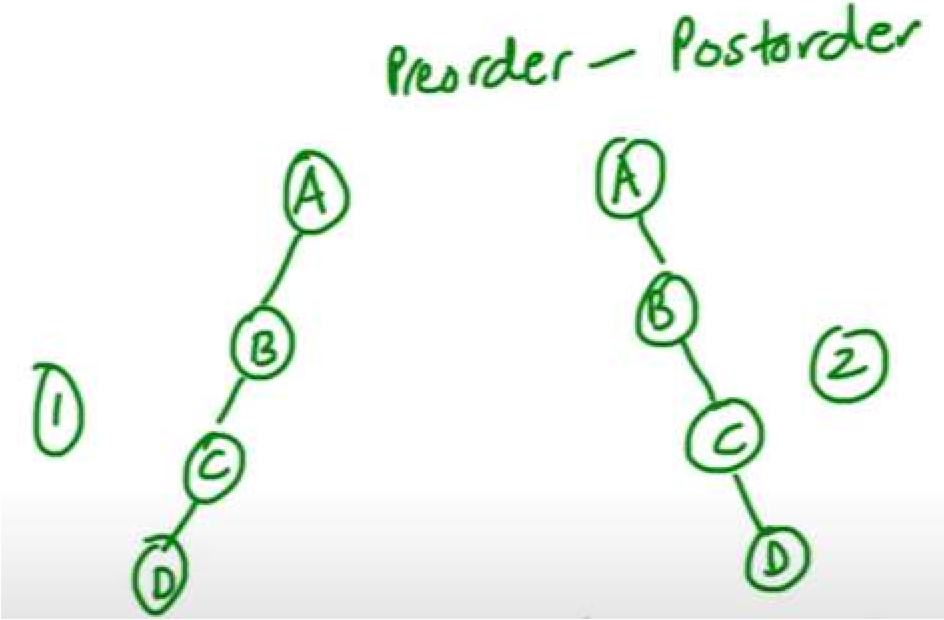
The same approach is used for the construction of **binary search tree**.

Complexity for the construction of BST = O(nlogn)

Complexity for the construction of Binary Tree = O(n^2)

Construct Binary Tree from Preorder and Postorder Traversal

- Cannot construct a unique binary tree
- Can construct a unique full binary tree



Preorder: A B C D
Post order: D C B A

Contd…

- Pre-order: 1, 2, 4, 8, 9, 5, 3, 6, 7 (Root->Left->Right)
- Post-order: 8, 9, 4, 5, 2, 6, 7, 3, 1 (Left->Right->Root)

```
          1
        /   \
      2       3
    /   \   /   \
   4     5 6     7
  / \
 8   9
```

Number of trees with n keys

- n = 3 {1, 2, 3}

How many BST can you draw?

5 BST we can draw.

- n= 4

Number of BST ??

- For n

Generalized formula???