COMP 5481

Programming and Problem Solving

Binary Trees Problem

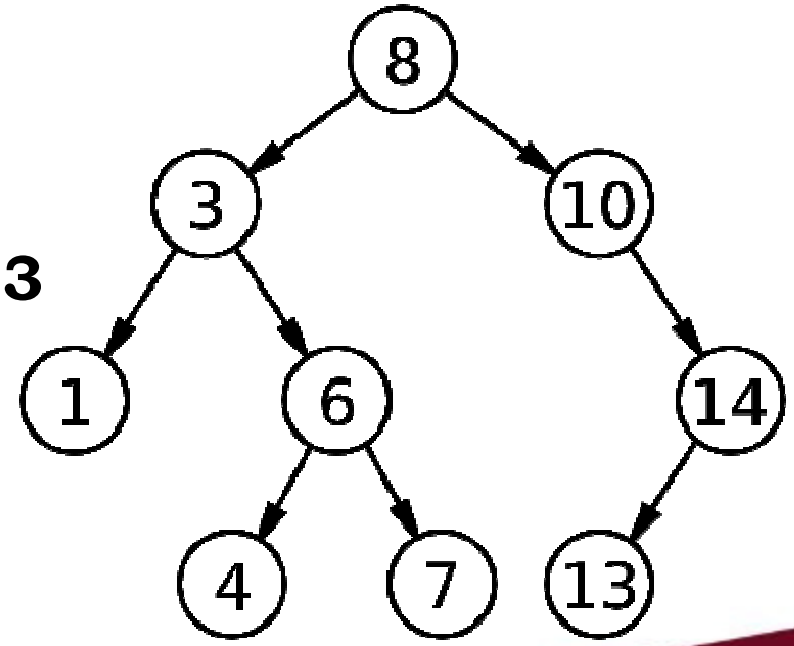
Binary Trees

Problem Solving Approach

- **Problem Statement: Print preorder traversal of a BST when postorder traversal is given as an input to the algorithm.**

Input: 1 4 7 6 3 13 14 10 8

Output: 8 3 1 6 4 7 10 14 13



Binary Trees

```
class Node
{
        int element; // Value stored in node
        Node left, right; // Left and right child
        Node(int val)
        {
            element = val;
            left = null; right = null;
        }
}
```

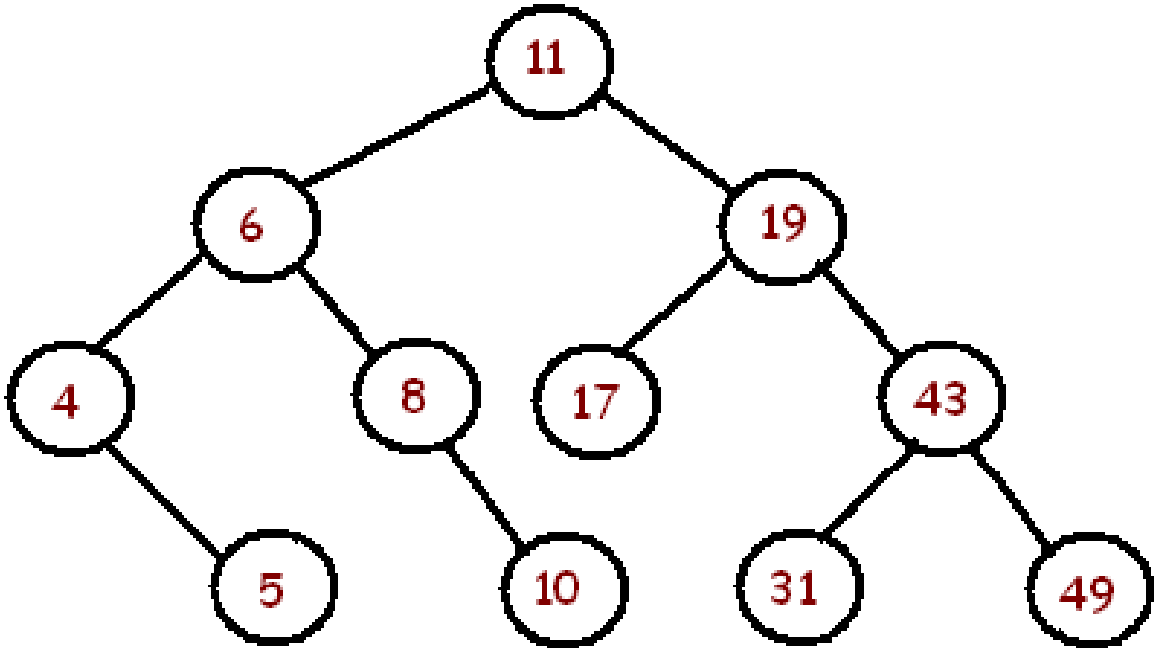
Binary Trees

```
class BinaryTree
{
      Node constructTree(int post[], int size)
      {
       }
      Node helperfunction()
      {
      }
     // A function to print preorder traversal
    void printPreorder(Node node)
    {
        if (node == null)
            return;
        System.out.print(node.data + " ");
        printPreorder(node.left);
        printPreorder(node.right);
    }
}
```

Binary Trees

```
public static void main(String[] args)
{
        BinaryTree tree = new BinaryTree();
        int post[] = new int[]{1, 4, 7, 6, 3, 13, 14, 10, 8};
        int size = post.length;
        Node root = tree.constructTree(post, size);
        tree.printPreorder(root);
}
```

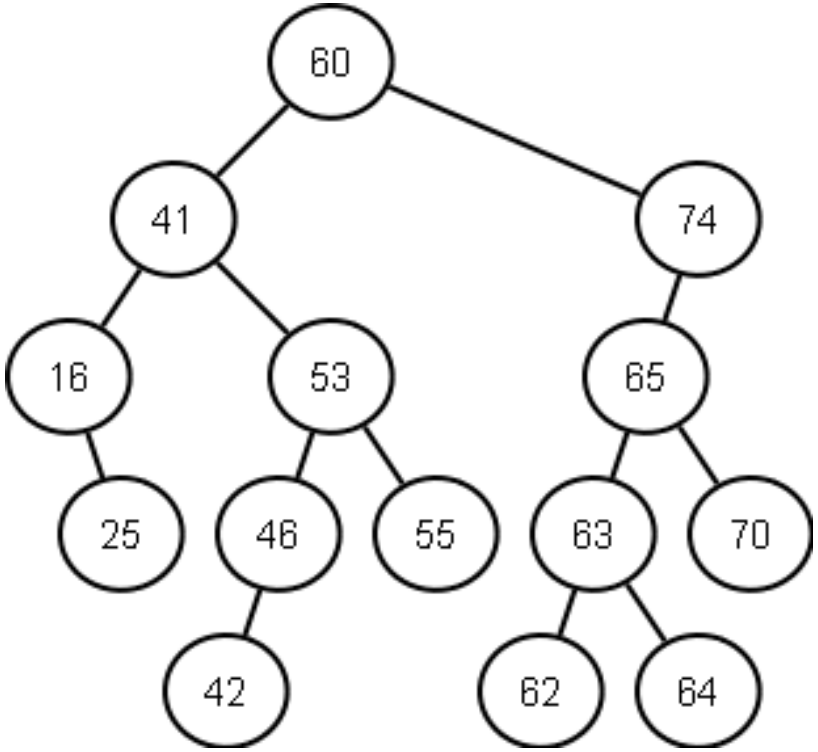
Binary Trees



Input: 5 4 10 8 6 17 31 49 43 19 11

Output: 11 6 4 5 8 10 19 17 43 31 49

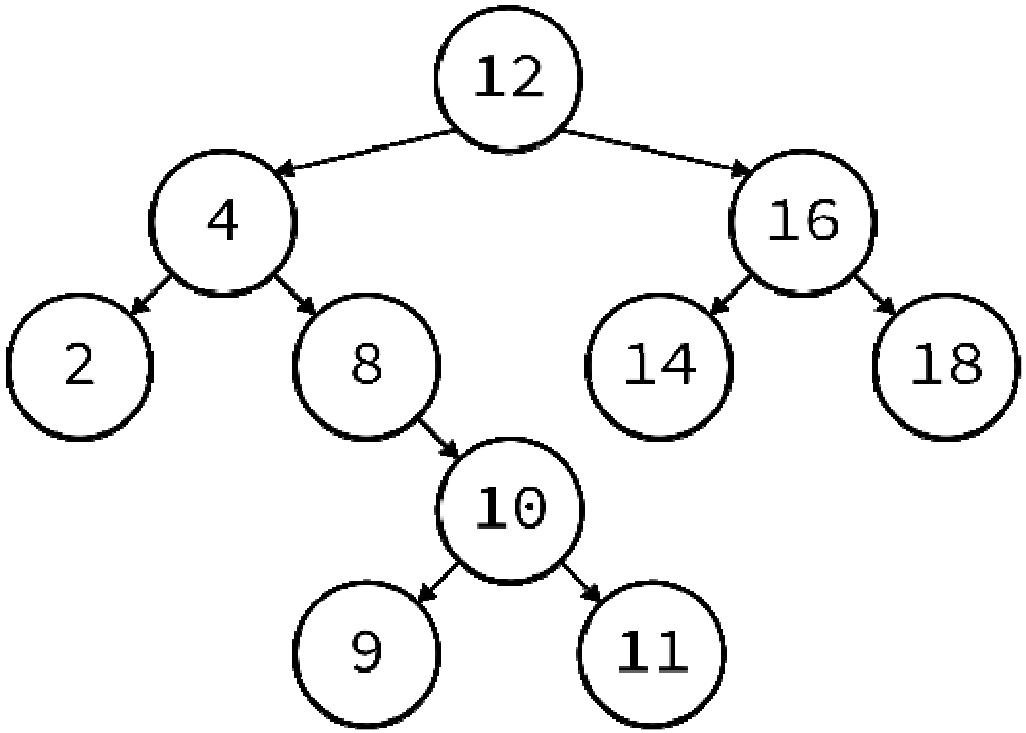
Binary Trees



Input: 25 16 42 46 55 53 41 62 64 63 70 65 74 60

Output: 60 41 16 25 53 46 42 55 74 65 63 62 64 70

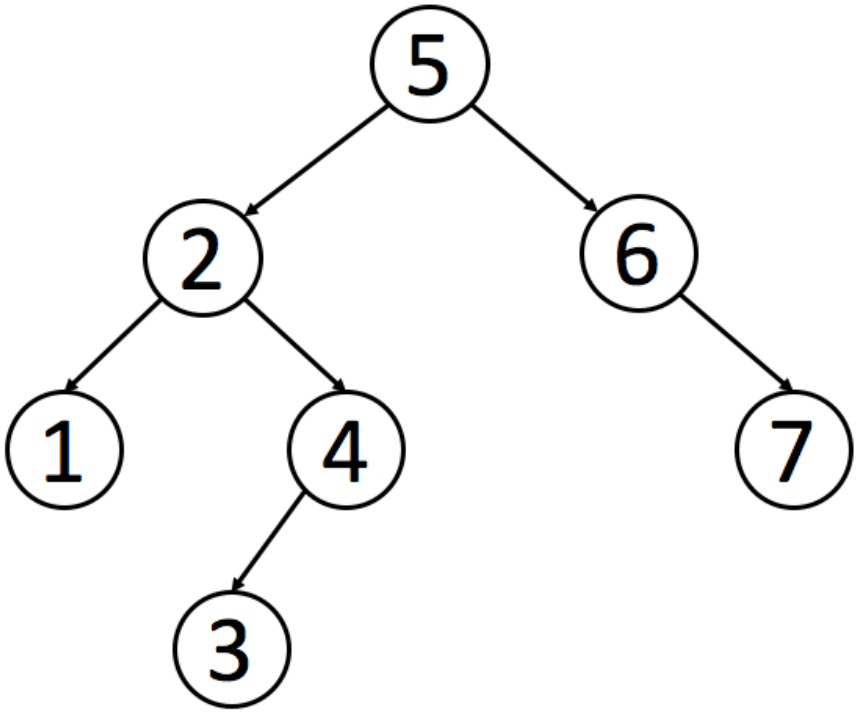
Binary Trees



Input: 2 9 11 10 8 4 14 18 16 12

Output: 12 4 2 8 10 9 11 16 14 18

Binary Trees



Input: 1 3 4 2 7 6 5

Output: 5 2 1 4 3 6 7