Lec-40

08/08/25 7:20 PM

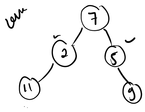
Tree → Assignment → input → level-order

7 2 5 11 9

Level-order

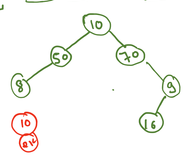
pre-✓ in-✓ ✗

-1 → null



[10 50 70 8 -1 -1 9 -1 -1 16 -1 -1 -1]

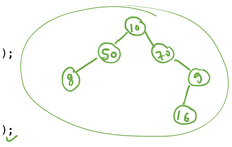
[10 50 70 8 -1 -1 9 -1 -1 16 -1 -1 -1] → input → Tree



```
private void Create_Tree() {
    // TODO Auto-generated method stub
    Scanner sc = new Scanner(System.in);
    Queue<Node> q = new LinkedList<>();
    int item = sc.nextInt();
    Node nn = new Node();
    nn.val = item;
    root = nn;
    q.add(nn);
}
```

```
private void Create_Tree() {
    // TODO Auto-generated method stub
    Scanner sc = new Scanner(System.in);
    Queue<Node> q = new LinkedList<>();
    int item = sc.nextInt();
    Node nn = new Node();
    nn.val = item;
    root = nn;
    q.add(nn);
    while(!q.isEmpty()) {
        Node r=q.poll();
        int c1=sc.nextInt();
        int c2=sc.nextInt();
        if(c1!=-1) {
            Node node = new Node();
            node.val=c1;
            r.left=node;
            q.add(node);
        }
        if(c2!=-1) {
            Node node = new Node();
            node.val=c2;
            r.right=node;
            q.add(node);
        }
    }
}
```

[10 50 70 8 -1 -1 9 -1 -1 16 -1 -1 -1] → input → Tree



Trace

Input: preorder = [3,9,20,15,7], inorder = [9,3,15,20,7]
Output: [3,9,20,null,null,15,7]

Root left right

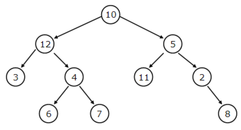
Left Root right

(i) root → pre-order

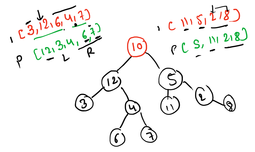
(ii) val → in-order

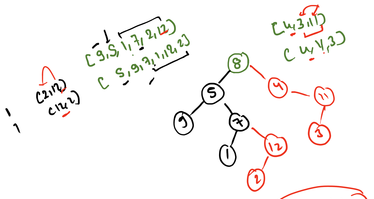
Left

Right

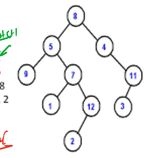


Levelorder tree traversal 10, 12, 5, 3, 4, 11, 2, 6, 7, 8
Inorder tree traversal 3, 12, 6, 4, 7, 10, 11, 5, 2, 8
Preorder tree traversal 10, 12, 3, 4, 6, 7, 5, 11, 2, 8
Postorder tree traversal 3, 6, 7, 4, 12, 11, 8, 2, 5, 10



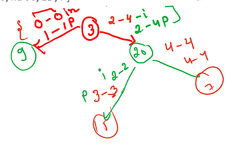


PreOrder - 8, 5, 9, 7, 1, 12, 2, 4, 11, 3
InOrder - 9, 5, 1, 7, 2, 12, 8, 4, 3, 11
PostOrder - 9, 1, 2, 12, 7, 5, 3, 11, 4, 8
LevelOrder - 8, 5, 4, 9, 7, 11, 1, 12, 3, 2



```
public TreeNode CreateTree(int[] pre, int[] in, int ilo, int ihi, int plo, int phi) {
    if (ilo > ihi || plo > phi) {
        return null;
    }
    TreeNode node = new TreeNode(pre[plo]);
    int idx = search(in, ilo, ihi, pre[plo]);
    int c = idx - ilo;
    node.left = CreateTree(pre, in, ilo, idx - 1, plo + 1, plo + c);
    node.right = CreateTree(pre, in, idx + 1, ihi, plo + c + 1, phi);
    return node;
}
```

Input: preorder = [3,9,20,15,7], inorder = [9,3,15,20,7]
Output: [3,9,20,null,null,15,7]



(1) Abstract

Class

Method

Pair

Collections

Interface

Thread

Stream