PostOrder

class Solution {

public List<Integer> postorderTraversal(TreeNode root) {

List<Integer> ans = new ArrayList<Integer>();

if(root == null) {

return ans;

}

Stack<TreeNode> stack = new Stack<TreeNode>();

// We will have a pointer to the recently popped node

TreeNode curr = root, prev = null;

while(curr != null || !stack.isEmpty()) {

// Keep on iterating towards the leftmost node

while(curr != null) {

stack.push(curr);

curr = curr.left;

}

// If there is no right child

// or right child is the one that we recently visited

// it means we have traversed all the nodes of stack.peek()

if(stack.peek().right == null || stack.peek().right == prev) {

// we will update the prev node

prev = stack.pop();

ans.add(prev.val);

} else {

// Otherwise we will visit the right child.

curr = stack.peek().right;

}

}

return ans;

}

}