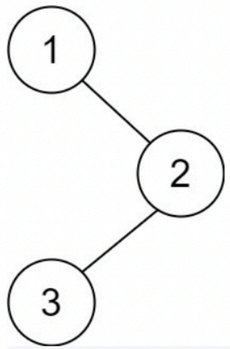
Description | Solution | Discuss (999+) | Submissions

145. Binary Tree Postorder Traversal

Easy 3633 133 Add to List Share

Given the `root` of a binary tree, return *the postorder traversal of its nodes' values*.

Example 1:




```
Input: root = [1,null,2,3]
Output: [3,2,1]
```

Example 2:

```
Input: root = []
Output: []
```

Example 3:

```
Input: root = [1]
Output: [1]
```

Java | Autocomplete

```java
class Solution {
    public List<Integer> postorderTraversal(TreeNode root) {
        List<Integer> results = new ArrayList<>();
        Stack<TreeNode> stack1 = new Stack<>();
        Stack<TreeNode> stack2 = new Stack<>();

        if(root == null) return results;

        stack1.push(root);

        while(!stack1.isEmpty()) {
            TreeNode node = stack1.pop();
            stack2.push(node);

            if(node.left != null) {
                stack1.push(node.left);
            }
            if(node.right != null) {
                stack1.push(node.right);
            }
        }

        while(!stack2.isEmpty()) {
            results.add(stack2.pop().val);
        }

        return results;
```

Testcase | Run Code Result | Debugger

Accepted Runtime: 0 ms

Your input	[1,null,2,3]
Output	[3,2,1]
Expected	[3,2,1]

Diff