

[Description](#)
[Solution](#)
[Discuss \(999+\)](#)
[Submissions](#)
*i* C++

## 145. Binary Tree Postorder Traversal

**Easy**

4552

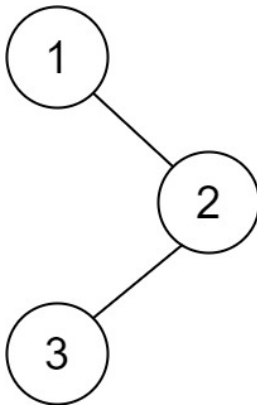
144

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]`

### Constraints:

- The number of the nodes in the tree is in the range `[0, 100]`.
- `-100 <= Node.val <= 100`

**Follow up:** Recursive solution is trivial, could you do it iteratively?

```

1  /
2  * Definition of a binary tree node.
3  * struct TreeNode {
4  *     int val;
5  *     TreeNode *left;
6  *     TreeNode *right;
7  *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
8  *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
9  *     ~TreeNode() {}
10 * };
11 */
12 class Solution {
13 public:
14     vector<int> postorderTraversal(TreeNode* root) {
15         vector<int> ans;
16         postorderTraversal(root, ans);
17         return ans;
18     }
19 private:
20     void postorderTraversal(TreeNode* root, vector<int> &ans) {
21         if (!root) return;
22         postorderTraversal(root->left, ans);
23         postorderTraversal(root->right, ans);
24         ans.push_back(root->val);
25     }
26 };

```

[Testcase](#)
[Run Code Re](#)
**Accepted**

Runti

 Your input `[1,`

 Output `[3,`

 Expected `[3,`

 Console [Use](#)

Run Code

Problems

Pick One

Prev

145/2344

Next