

144. Binary Tree Preorder Traversal ★

[Question](#)[Editorial Solution](#)[My Submissions \(/problems/binary-tree-preorder-traversal/submissions/\)](#)

Total Accepted: **143810** Total Submissions: **342744** Difficulty: **Medium**

Given a binary tree, return the *preorder* traversal of its nodes' values.

For example:

Given binary tree {1,#,2,3} ,

```
1
 \
  2
 /
3
```

return [1,2,3] .

Note: Recursive solution is trivial, could you do it iteratively?

[Subscribe \(/subscribe/\)](#) to see which companies asked this question

[Show Tags](#)[Show Similar Problems](#)

Have you met this question in a real interview?

[Discuss \(https://leetcode.com/discuss/questions/oj/binary-tree-preorder-traversal\)](https://leetcode.com/discuss/questions/oj/binary-tree-preorder-traversal)[Pick One \(/problems/random-one-question/\)](#)[Notes](#)

C++



</>

```
1 /**
2  * Definition for a binary tree node.
3  * struct TreeNode {
4  *     int val;
5  *     TreeNode *left;
6  *     TreeNode *right;
7  *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
8  * };
9  */
10 class Solution {
11 public:
12     vector<int> preorderTraversal(TreeNode* root) {
13         vector<int> visit;
14         if (root) {
```

```
15         std::stack<TreeNode*> stack;
16         TreeNode* now = root;
17         while (now) {
18             if (now->right) stack.push(now->right);
19             visit.push_back(now->val);
20             now = now->left;
21             if (now == NULL && !stack.empty()) {
22                 now = stack.top();
23                 stack.pop();
24             }
25         }
26     }
27     return visit;
28 }
29 };
```

Custom Testcase ☐

Run Code

Submit Solution

[Frequently Asked Questions \(/faq/\)](/faq/) | [Terms of Service \(/tos/\)](/tos/)

[Privacy](#)

✉ Send Feedback (<mailto:admin@leetcode.com?subject=Feedback>)
Copyright © 2016 LeetCode

📄 Notes