

Alex Vargas A00824525

# Reto Semestral Pt.- 2

## Problema 1 (Easy)

Practice

Data Structures

Trees

Tree: Postorder Traversal

Exit Full Screen View

Problem

Submissions

Leaderboard

Discussions

Editorial

Complete the postOrder function in your editor below, which has 1 parameter: a pointer to the root of a binary tree. It must print the values in the tree's postorder traversal as a single line of space-separated values.

Input Format

Our hidden tester code passes the root node of a binary tree to your postOrder function.

Constraints

1 ≤ Nodes in the tree ≤ 500

Output Format

Print the tree's postorder traversal as a single line of space-separated values.

Sample Input

```
1
 /
2
 / \
5 6
 / \
3 4
```

Sample Output

4 3 6 5 2 1

Change Theme

C++

```
1 > #include <bits/stdc++.h>
35
36 /* you only have to complete the function given below.
37 Node is defined as
38
39 class Node {
40 public:
41     int data;
42     Node *left;
43     Node *right;
44     Node(int d) {
45         data = d;
46         left = NULL;
47         right = NULL;
48     }
49 };
50
51 */
52
53 void postOrder(Node *root) {
54     Node *curr = root;
55     if (root != NULL){
56         postOrder(curr->left);
57         postOrder(curr->right);
58         cout<<curr->data<<" ";
59     }
60 }
61
62
63 > }; //End of Solution
```

Line: 35 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code

Problem

Submissions

Leaderboard

Discussions

Editorial

Complete the postOrder function in your editor below, which has 1 parameter: a pointer to the root of a binary tree. It must print the values in the tree's postorder traversal as a single line of space-separated values.

Input Format

Our hidden tester code passes the root node of a binary tree to your postOrder function.

Constraints

1 ≤ Nodes in the tree ≤ 500

Output Format

Print the tree's postorder traversal as a single line of space-separated values.

Sample Input

```
1
 /
2
 / \
5 6
 / \
3 4
```

Sample Output

4 3 6 5 2 1

Line: 35 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code

CONGRATULATIONS

You have earned 50,000 points!

You are now 34 points away from the 3rd star for your problem solving badge!

30%

Congratulations

You solved this challenge. Would you like to challenge your friends?

Facebook Twitter Email

Next Challenge

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Complex Message

Success

Input (stdin)

4

1 2 3 4 5 6

Expected Output

4 3 6 5 2 1

### Problema 2 (Easy)

Problem

Submissions

Leaderboard

Discussions

Editorial

Complete the preOrder function in your editor below, which has 1 parameter: a pointer to the root of a binary tree. It must print the values in the tree's preorder traversal as a single line of space-separated values.

**Input Format**

Our hidden tester code passes the root node of a binary tree to your preOrder function.

**Constraints**

$1 \leq$  Nodes in the tree  $\leq 500$

**Output Format**

Print the tree's preorder traversal as a single line of space-separated values.

**Sample Input**

```
1
 \
  2
   \
    5
   / \
  3   6
   \
    4
```

**Sample Output**

```
1 2 5 3 4 6
```

Change Theme

C++

Line: 35 Col: 1

```
> #include <bits/stdc++.h>
35
36 /* you only have to complete the function given below.
37 Node is defined as
38
39 class Node {
40     public:
41         int data;
42         Node *left;
43         Node *right;
44         Node(int d) {
45             data = d;
46             left = NULL;
47             right = NULL;
48         }
49 };
50
51
52
53
54
55
56
57
58
59
60
61 >}; //End of Solution=
```

Upload Code as File

Test against custom input

Run Code

Submit Code

Problem

Submissions

Leaderboard

Discussions

Editorial

Complete the preOrder function in your editor below, which has 1 parameter: a pointer to the root of a binary tree. It must print the values in the tree's preorder traversal as a single line of space-separated values.

**Input Format**

Our hidden tester code passes the root node of a binary tree to your preOrder function.

**Constraints**

$1 \leq$  Nodes in the tree  $\leq 500$

**Output Format**

Print the tree's preorder traversal as a single line of space-separated values.

**Sample Input**

```
1
 \
  2
   \
    5
   / \
  3   6
   \
    4
```

**Sample Output**

```
1 2 5 3 4 6
```

Upload Code as File

Test against custom input

Run Code

Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

[f](#) [t](#) [m](#)

Next Challenge

Test case 0

Computer Message

Success

Test output

Download

1 2 5 3 4 6

Expected Output

Download

1 2 5 3 4 6

## Problema 3 (Medium)

Description

Solution

Submissions

Discuss (999+)

C++

Autocomplete

### 94. Binary Tree Inorder Traversal

Medium 2629 115 Add to List Share

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

**Example:**

```

Input: [1,null,2,3]
      1
       \
        2
       / \
      3  4
Output: [1,3,2]
    
```

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

Accepted 658,183 Submissions 1,072,294

Seen this question in a real interview before?

Contributor

Companies

Related Topics

Similar Questions

```

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
11 class Solution {
12 public:
13     vector<int> inorderTraversal(TreeNode* root) {
14         vector<int> v1;
15         inorder(root, v1);
16         return v1;
17     }
18
19     void inorder(TreeNode* root, vector<int> &v){
20         if (root != NULL){
21             inorder(root->left, v);
22             v.push_back(root->val);
23             inorder(root->right, v);
24         }
25     }
26 };
    
```

Testcase

Run Code Result

Debugger

**Accepted** Runtime: 4 ms

Your input [1,null,2,3]

Output [1,3,2]

Expected [1,3,2]

Problems

Pick One

< Prev

1/236

Next >

Console

How to create a testcase

Run Code

Submit

### Success Details >

Runtime: **0 ms**, faster than **100.00%** of C++ online submissions for Binary Tree Inorder Traversal.

Memory Usage: **6.9 MB**, less than **100.00%** of C++ online submissions for Binary Tree Inorder Traversal.

Next challenges:

- Validate Binary Search Tree
- Binary Tree Preorder Traversal
- Binary Tree Postorder Traversal
- Binary Search Tree Iterator
- Kth Smallest Element in a BST
- Closest Binary Search Tree Value II
- Inorder Successor in BST
- Convert Binary Search Tree to Sorted Doubly Linked List
- Minimum Distance Between BST Nodes

Show off your acceptance:



Time Submitted	Status	Runtime	Memory	Language
a few seconds ago	Accepted	0 ms	6.9 MB	cpp

Problema 4 (Medium)

LeetCode

ExploreProblemsMockContestArticlesDiscussStore

30-Day LeetCode ChallengePremium

LeetCode

DescriptionSolutionSubmissionsDiscuss (499)

C++Autocomplete

814. Binary Tree Pruning

Medium86233Add to ListShare

We are given the head node `root` of a binary tree, where additionally every node's value is either a 0 or a 1.

Return the same tree where every subtree (of the given tree) not containing a 1 has been removed.

(Recall that the subtree of a node `X` is `X`, plus every node that is a descendant of `X`.)

Example 1:

Input: [1,null,0,0,1]

Output: [1,null,0,null,1]

Explanation:

Only the red nodes satisfy the property "every subtree not containing a 1".

The diagram on the right represents the answer.

Example 2:

Input: [1,0,1,0,0,0,1]

Output: [1,null,1,null,1]

Example 3:

Input: [1,1,0,1,1,0,1,0]

Output: [1,1,0,1,1,null,1]

Problems

Pick One

< Prev

8/236

Next >

Console

Contribute

Run Code

Submit

```
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
11
12 class Solution {
13 public:
14     TreeNode* pruneTree(TreeNode* root) {
15         if (root == NULL) return NULL;
16         root->left = pruneTree(root->left);
17         root->right = pruneTree(root->right);
18         if (root->left == NULL && root->right == NULL && root->val == 0){
19             return NULL;
20         }else{
21             return root;
22         }
23     }
24 };
25
26
```

Your previous code was restored from your local storage. [Reset to default](#)




Success Details >

Runtime: 0 ms, faster than 100.00% of C++ online submissions for Binary Tree Pruning.

Memory Usage: 9 MB, less than 100.00% of C++ online submissions for Binary Tree Pruning.

Next challenges:

- Maximum Depth of Binary Tree
- Construct Binary Tree from Preorder and Postorder Traversal
- Univalued Binary Tree

Show off your acceptance:   

Time Submitted	Status	Runtime	Memory	Language
a few seconds ago	Accepted	0 ms	9 MB	cpp

Problema 5 (Hard)

DescriptionSolutionSubmissionsDiscuss (999+)

145. Binary Tree Postorder Traversal

Hard145676Add to ListShare

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

Example:

Input: [1,null,2,3]

1

\

2

/

3

Output: [3,2,1]

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

Accepted 345,347Submissions 651,140

Seen this question in a real interview before?

YesNo

Contributor

Companies

Related Topics

Similar Questions

C++Autocomplete

```
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
11 class Solution {
12 public:
13     vector<int> postorderTraversal(TreeNode* root) {
14         vector<int> v;
15         TreeNode *curr = root;
16         postOrder(curr,v);
17         return v;
18     }
19
20     void postOrder(TreeNode* root, vector<int> &v){
21         if (root != NULL){
22             postOrder(root->left,v);
23             postOrder(root->right,v);
24             v.push_back(root->val);
25         }
26     }
27
28 };
```

TestcaseRun Code ResultDebugger

AcceptedRuntime: 0 ms

Your input[1,null,2,3]

Output[3,2,1]Diff

Expected[3,2,1]

ConsoleHow to create a testcase

Run CodeSubmit

SuccessDetails

Runtime: 0 ms, faster than 100.00% of C++ online submissions for Binary Tree Postorder Traversal.

Memory Usage: 7.1 MB, less than 100.00% of C++ online submissions for Binary Tree Postorder Traversal.

Next challenges:

N-ary Tree Postorder Traversal

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
a few seconds ago	Accepted	0 ms	7.1 MB	cpp