

GitHub ClassroomCU-Assignments/assignment4-Binary Tree Level Order Travers...

leetcode.com/problems/binary-tree-level-order-traversal/submissions/1549532421/

Problem ListRunSubmit

DescriptionAcceptedEditorialSolutionsSubmissions

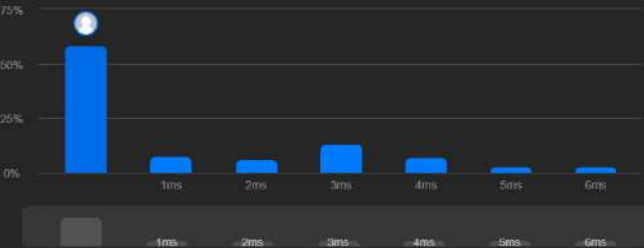
All Submissions

Accepted35 / 35 test cases passed
Tarush03 submitted at Feb 20, 2025 15:59

Runtime
0 ms | Beats 100.00%

Memory
17.09 MB | Beats 71.00%

Analyze Complexity



Time Interval	Percentage
0ms	~65%
1ms	~5%
2ms	~5%
3ms	~10%
4ms	~5%
5ms	~2%
6ms	~2%

CodeC++

```
class Solution {
public:
    vector<vector<int>> levelOrder(TreeNode* root) {
        vector<vector<int>> ans;
        if(root==NULL) return ans;
        queue<TreeNode*> q;
        q.push(root);
        while(!q.empty()){
            int s=q.size();
            vector<int> v;
            for(int i=0;i<s;i++){
                TreeNode* node=q.front();
                q.pop();
```

Code

C++Auto

```
1 class Solution {
2 public:
3     vector<vector<int>> levelOrder(TreeNode* root) {
4         vector<vector<int>> ans;
5         if(root==NULL) return ans;
6         queue<TreeNode*> q;
7         q.push(root);
8         while(!q.empty()){
9             int s=q.size();
10            vector<int> v;
11            for(int i=0;i<s;i++){
12                TreeNode* node=q.front();
13                q.pop();
```

SavedLn 1, Col 1

Continue to work on your code from 2/20/2025, 3:58:58 PMRestore

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2Case 3

Input

root =
[3,9,20,null,null,15,7]

Output

[[3],[9,20],[15,7]]

GitHub ClassroomCU-Assignments/assignment4-Binary Tree Inorder Traversal

leetcode.com/problems/binary-tree-inorder-traversal/submissions/1549533251/

Problem ListRunSubmit

AcceptedSubmissions

Accepted71 / 71 test cases passed

Tarush03 submitted at Feb 20, 2025 16:00

EditorialSolution

Runtime0 ms | Beats 100.00%

Analyze Complexity

Memory11.03 MB | Beats 15.35%

Runtime	Beats
0 ms	100.00%
1 ms	0.00%
2 ms	0.00%
3 ms	0.00%
4 ms	0.00%

CodeC++

```
class Solution {
public:
    vector<int> ans;
    void helper(TreeNode* root){
        if(!root) return;
        helper(root->left);
        ans.push_back(root->val);
        helper(root->right);
    }
    vector<int> inorderTraversal(TreeNode* root) {
        helper(root);
        return ans;
    }
};
```

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2Case 3Case 4

Input

root = [1,null,2,3]

Output

[1,3,2]

GitHub ClassroomCU-Assignments/assignment4-Binary Tree Zigzag Level Order

leetcode.com/problems/binary-tree-zigzag-level-order-traversal/

Problem ListRunSubmit

DescriptionEditorialSolutionsAcceptedSubmissions

All Submissions

Accepted33 / 33 test cases passed
Tarush03 submitted at Feb 20, 2025 16:02

EditorialSolution

Runtime0 ms | Beats 100.00%
Analyze Complexity

Memory15.25 MB | Beats 18.17%

Runtime	1ms	2ms	3ms	4ms
Percentage	100%	~0%	~0%	~0%

CodeC++

```
class Solution {
public:
    vector<vector<int>> zigzagLevelOrder(TreeNode* root) {
        if (!root) return {};
        vector<vector<int>> result;
        queue<TreeNode*> q;
        q.push(root);
        bool leftToRight = true;
```

Code

```
15 q.pop();
16 int index = leftToRight ? i : (levelSize - 1 - i);
17 level[index] = node->val;
18 if (node->left) q.push(node->left);
19 if (node->right) q.push(node->right);
20 }
21 leftToRight = !leftToRight;
22 result.push_back(level);
23 }
24 return result;
25 }
26 ;
```

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2Case 3

Input

root =
[3,9,20,null,null,15,7]

Output

[[3],[20,9],[15,7]]

GitHub Classroom x CU-Assignments/assignment4- x Construct Binary Tree from Preorder and Inorder Traversal

leetcode.com/problems/construct-binary-tree-from-preorder-and-inorder-traversal/submissions/1549535229/

Problem List Run Submit Premium

Description Editorial Solutions Accepted Submissions

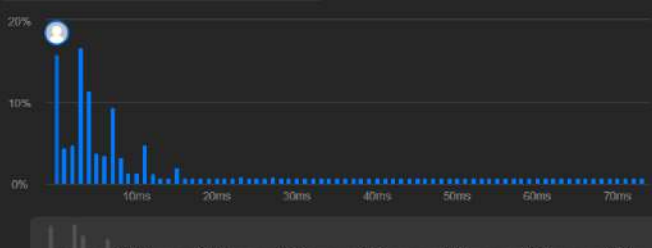
All Submissions

Accepted 203 / 203 testcases passed

Tarush03 submitted at Feb 20, 2025 16:03

Runtime 0 ms | Beats 100.00% Memory 27.51 MB | Beats 39.21%

Analyze Complexity



Code C++

```
class Solution {
    unordered_map<int, int> inorderMap;
    TreeNode* buildTreeHelper(vector<int>& preorder, int preStart, int preEnd,
                             vector<int>& inorder, int inStart, int inEnd) {
        if (preStart > preEnd || inStart > inEnd) return nullptr;
        TreeNode* root = new TreeNode(preorder[preStart]);
        int rootIndex = inorderMap[root->val];
        int leftSize = rootIndex - inStart;

        root->left = buildTreeHelper(preorder, preStart + 1, preStart + leftSize,
                                    inorder, inStart, rootIndex - 1);
        root->right = buildTreeHelper(preorder, preStart + leftSize + 1, preEnd,
                                      inorder, rootIndex + 1, inEnd);

        return root;
    }
};
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

preorder = [3, 9, 20, 15, 7]

inorder = [9, 3, 15, 20, 7]

GitHub Classroom CU-Assignments/assignment4- Path Sum II - LeetCode

leetcode.com/problems/path-sum-ii/submissions/1549536692/

Problem List < > 🔍

Description Editorial Solutions Accepted Submissions

All Submissions


Accepted 115 / 115 testcases passed

Tarush03 submitted at Feb 20, 2025 16:05

Editorial Solution

Runtime 0 ms Beats 100.00% Memory 20.88 MB Beats 89.44%

Analyze Complexity



Code C++

```
class Solution {
public:
    vector<vector<int>> pathSum(TreeNode* root, int sum) {
```

Code

```
private:
void findPaths(TreeNode* node, int sum, vector<int>& path, vector<vector<int>>& paths) {
    if (!node) return;
    path.push_back(node->val);
    if (!(node->left) && !(node->right) && sum == node->val)
        paths.push_back(path);
    findPaths(node->left, sum - node->val, path, paths);
    findPaths(node->right, sum - node->val, path, paths);
    path.pop_back();
}
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

root =

[5,4,8,11,null,13,4,7,2,null,null,5,1]

GitHub Classroom CU-Assignments/assignment4- Flatten Binary Tree to Linked List

leetcode.com/problems/flatten-binary-tree-to-linked-list/

Problem List Run Submit

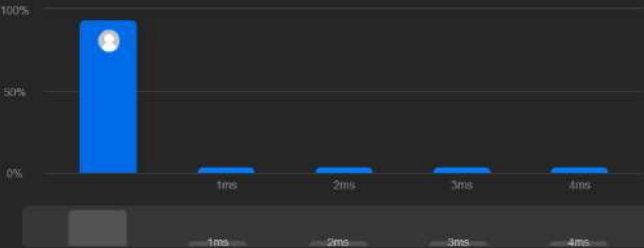
Description Editorial Solutions Accepted Submissions

All Submissions

Accepted 225 / 225 testcases passed
Tarush03 submitted at Feb 20, 2025 16:07

Runtime 0 ms | Beats 100.00%
Memory 17.62 MB | Beats 22.75%

Analyze Complexity



Code C++

```
class Solution {
private:
    TreeNode* prev = nullptr;
public:
    void flatten(TreeNode* root) {
        if (!root) return;
        flatten(root->right);
        flatten(root->left);
        root->right=prev;
        root->left=nullptr;
        prev=root;
    }
};
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

root =
[1,2,5,3,4,null,6]

Output

[1,null,2,null,3,null,4,null,5,null,6]

GitHub ClassroomCU-Assignments/assignment4-Binary Tree Preorder Traversal

leetcode.com/problems/binary-tree-preorder-traversal/submissions/1549538653/

Problem ListRunSubmit

AcceptedSubmissions

Accepted71 / 71 testcases passed


Tarush03 submitted at Feb 20, 2025 16:08

EditorialSolution

Runtime0 ms | Beats 100.00%

Analyze Complexity

Memory11.07 MB | Beats 16.18%



Runtime	Beats
0 ms	100.00%
1 ms	~0%
2 ms	~0%
3 ms	~0%
4 ms	~0%

CodeC++Auto

```
1 class Solution {
2 public:
3     vector<int> preorderTraversal(TreeNode* root) {
4         vector<int> preorder;
5         stack<TreeNode*> stack;
6         if (root == NULL)
7             return preorder;
8         stack.push(root);
9         while(!stack.empty()) {
10             TreeNode* curr = stack.top();
11             stack.pop();
12             preorder.push_back(curr->val);
13             if (curr->right != NULL)
14                 stack.push(curr->right);
15             if (curr->left != NULL)
```

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2Case 3Case 4

Input

root =
[1,null,2,3]

Output

[1,2,3]

CodeC++

```
class Solution {
public:
    vector<int> preorderTraversal(TreeNode* root) {
        vector<int> preorder;
        stack<TreeNode*> stack;
        if (root == NULL)
            return preorder;
        stack.push(root);
```

GitHub ClassroomCU-Assignments/assignment4-1Lowest Common Ancestor of a Binary Tree

leetcode.com/problems/lowest-common-ancestor-of-a-binary-tree/submissions/1549540018/

Problem ListRunSubmit

Accepted

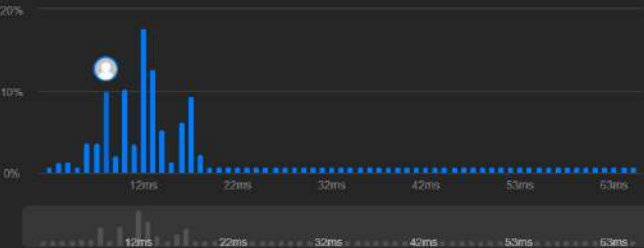
All Submissions

Accepted32 / 32 testcases passed

Tarush03 submitted at Feb 20, 2025 16:10

Runtime8 ms | Beats 89.60%

Memory17.44 MB | Beats 40.93%



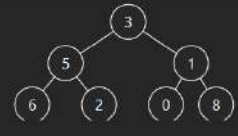
Code | C++

```
#include <iostream>
using namespace std;
class Solution {
public:
    TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q) {
        if (!root) return nullptr;
        if (root == p || root == q) return root;
        TreeNode* left = lowestCommonAncestor(root->left, p, q);
        TreeNode* right = lowestCommonAncestor(root->right, p, q);
        if (left && right) return root;
        return left ? left : right;
    }
};
```

Testcase | Test Result

Case 1Case 2Case 3

root = [3,5,1,6,2,0,8,null,null,7,4]



GitHub ClassroomCU-Assignments/assignment4-Binary Tree Cameras - LeetCode

leetcode.com/problems/binary-tree-cameras/submissions/1549541089/

Problem ListRunSubmit

DescriptionEditorialSolutionsAcceptedSubmissions

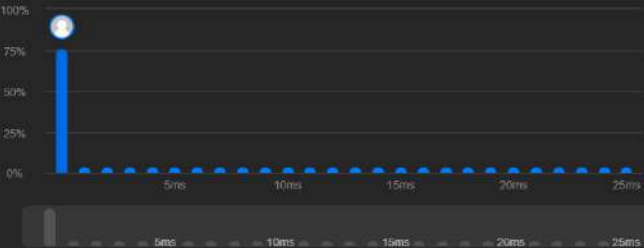
All Submissions

Accepted 171 / 171 testcases passed
Tarush03 submitted at Feb 20, 2025 16:11

EditorialSolution

Runtime0 ms | Beats 100.00%
Analyze Complexity

Memory21.50 MB | Beats 21.94%



100%
75%
50%
25%
0%

5ms10ms15ms20ms25ms

Code | C++

```
class Solution {
public:
    int solve(TreeNode* root, int &cameras) {
        if (!root) {
            return 1;
        }
        int left = solve(root->left, cameras);
        int right = solve(root->right, cameras);
```

Code

```
int left = solve(root->left, cameras);
int right = solve(root->right, cameras);
if (left == 0 || right == 0) {
    cameras++;
    return 2;
}
if (left == 2 || right == 2) {
    return 1;
}
return 0;
}

int minCameraCover(TreeNode* root) {
    int cameras = 0;
    int ans = solve(root, cameras);
    if (ans == 0) {
        ..
    }
}
```

TestcaseTest Result

Accepted Runtime: 0 ms

Case 1Case 2

Input

root =
[0,0,null,0,0]

Output

1

Vertical Order Traversal of a Bin

leetcode.com/problems/vertical-order-traversal-of-a-binary-tree/submissions/1549542090/

Problem List

Run

Submit

Premium

Description

Editorial

Solutions

Accepted

Submissions

All Submissions

Accepted 34 / 34 test cases passed

Tarush03 submitted at Feb 20, 2025 16:12

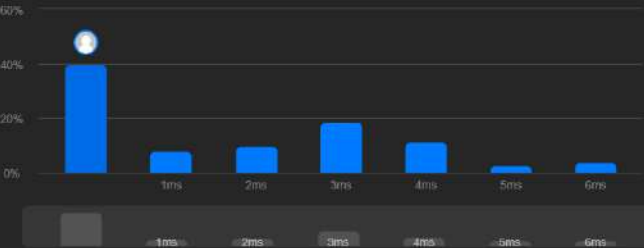
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

16.51 MB | Beats 17.91%



Runtime (ms)	Percentage (%)
0ms	~40%
1ms	~5%
2ms	~5%
3ms	~15%
4ms	~10%
5ms	~2%
6ms	~2%

Code

C++

Auto

```
22 int x = t.second.first, y = t.second.second;
23 nodes[x].insert(a->val);
24 if(a->left){
25     q.push({a->left, {x-1, y+1}});
26 }
27 if(a->right){
28     q.push({a->right, {x+1, y+1}});
29 }
30 }
31 }
32 vector<vector<int>>>ans;
33 for(auto p: nodes){
34     vector<int>col;
35     for(auto b:p.second){
36         col.insert(col.end(), b.second.begin(), b.second.end());
```

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

root = [3,9,20,null,null,15,7]

Output

[[9],[3,15],[20],[7]]

Code

C++

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 * }
```