

CU-Assignments/assignment5... x Validate Binary Search Tree - Le... x BST Validation C++ Solution x | +

leetcode.com/problems/validate-binary-search-tree/submissions/1569138595/

Gmail YouTube Maps Disney+ Hotstar... Tata Play All Bookmarks

Problem List < > < Run Submit < > Premium

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 86 / 86 testcases passed

MohitBehal submitted at Mar 10, 2025 19:11

Editorial Solution

Runtime 0 ms Beats 100.00% Analyze Complexity

Memory 21.95 MB Beats 48.51%

100% 50% 0% 1ms 2ms 3ms 4ms

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) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
 * };
 */
class Solution {
public:
    bool isValidBST(TreeNode* root, long long minVal = LLONG_MIN, long long maxVal = LLONG_MAX) {
        if (!root) return true;
        if (root->val <= minVal || root->val >= maxVal) return false;
        return isValidBST(root->left, minVal, root->val) &&
            isValidBST(root->right, root->val, maxVal);
    }
};
```

Saved Ln 19, Col 6

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

Very humid Now

Search

ENG IN 19:11 10-03-2025

Maximum Depth of Binary Tree

leetcode.com/problems/maximum-depth-of-binary-tree/submissions/1569133355/

Problem ListRunSubmit

DescriptionAcceptedEditorialSolutionsSubmissions

All Submissions

Accepted39 / 39 testcases passed

MohitBehal submitted at Mar 10, 2025 19:05

EditorialSolution

Runtime4 msBeats 1.38%

Memory19.07 MBBeats 45.21%

Analyze Complexity

| Runtime | Percentage |
|---------|------------|
| 1ms | 0% |
| 2ms | 0% |
| 3ms | 0% |
| 4ms | 100% |

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) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
 * };
 */
class Solution {
public:
    int maxDepth(TreeNode* root) {
        if(root==NULL) return 0;
        return 1+ max(maxDepth(root->left), maxDepth(root->right));
    }
};
```

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2

Input

23°C Mostly clear

Search

ENG IN

19:06 10-03-2025

CU-Assignments/assignment5...Sum of Left Leaves - LeetCode...BST Validation C++ Solution

leetcode.com/problems/sum-of-left-leaves/submissions/1569156159/

Problem List

RunSubmit

Premium

DescriptionAcceptedEditorialSolutionsSubmissions

All Submissions

Accepted100 / 100 testcases passed

MohitBehal submitted at Mar 10, 2025 19:30


Runtime

0 msBeats 100.00%

Memory

16.22 MBBeats 23.61%

Analyze Complexity



1ms2ms3ms4ms

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) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
 * }
```

Code

C++Auto

```
19 q.push(root);
20
21 while (!q.empty()) {
22     TreeNode* node = q.front();
23     q.pop();
24
25     if (node->left) {
26         if (!node->left->left && !node->left->right)
27             sum += node->left->val;
28         else
29             q.push(node->left);
30     }
31     if (node->right) {
32         q.push(node->right);
33     }
34 }
35
36 return sum;
37 }
```

SavedLn 36, Col 20

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2

Input

Air Severe Now

Search

19:30 10-03-2025

CU-Assignments/assignment5...Populating Next Right Pointers...BST Validation C++ Solution

leetcode.com/problems/populating-next-right-pointers-in-each-node/submissions/1569153343/

Problem List

RunSubmit

Premium

DescriptionAcceptedEditorialSolutionsSubmissions

All Submissions

Accepted59 / 59 testcases passed

MohitBehal submitted at Mar 10, 2025 19:27

Runtime

12 msBeats 69.87%

Analyze Complexity

Memory

18.82 MBBeats 90.91%

EditorialSolution

Runtime

12 msBeats 69.87%

Analyze Complexity

Memory

18.82 MBBeats 90.91%

EditorialSolution

Code | C++

```
// Definition for a Node.
class Node {
public:
    int val;
    Node* left;
    Node* right;
    Node* next;
    Node() {}
    Node(int _val) { val = _val; }
    Node(int _val, Node* _left, Node* _right, Node* _next) { val = _val; left = _left; right = _right; next = _next; }
};
```

Code

```
23
24
25
26 Node* leftMost = root;
27
28 while (leftMost->left) {
29     Node* curr = leftMost;
30
31     while (curr) {
32         curr->left->next = curr->right;
33         if (curr->next) {
34             curr->right->next = curr->next->left;
35         }
36         curr = curr->next;
37     }
38     leftMost = leftMost->left;
39 }
40
41 return root;
```

SavedLn 40, Col 21

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2

Input

22°CPartly cloudy

Search

ENG IN

19:2710-03-2025

CU-Assignments/assignment5...Kth Smallest Element in a BST - BST Validation C++ Solution

leetcode.com/problems/kth-smallest-element-in-a-bst/submissions/1569151017/

Problem ListRunSubmit

DescriptionAcceptedEditorialSolutionsSubmissions

All Submissions

Accepted93 / 93 testcases passedMohitBehal submitted at Mar 10, 2025 19:24

Runtime0 msBeats 100.00%

Memory24.36 MBBeats 67.94%

Analyze Complexity

100%

50%

0%

1ms2ms3ms4ms

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) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
 * }
 */
```

Code

C++Auto

```
14 int count = 0, result = 0;
15
16 void inorder(TreeNode* root, int k) {
17     if (!root) return;
18
19     inorder(root->left, k);
20     count++;
21     if (count == k) {
22         result = root->val;
23         return;
24     }
25     inorder(root->right, k);
26 }
27
28 int kthSmallest(TreeNode* root, int k) {
29     inorder(root, k);
30     return result;
31 }
32 ;
```

SavedLn 31, Col 6

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2

Input

23°C Mostly clear

Search

ENG IN

19:24 10-03-2025

CU-Assignments/assignment5... Binary Tree Level Order Travers... BST Validation C++ Solution

leetcode.com/problems/binary-tree-level-order-traversal/submissions/1569149186/

Problem List Run Submit

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 35 / 35 testcases passed

MohitBehal submitted at Mar 10, 2025 19:22

Runtime

0 ms Beats 100.00%

Analyze Complexity

Memory

17.23 MB Beats 16.86%

Bar chart showing runtime performance across different test cases. The first bar (0ms) is significantly higher than the others, indicating it is the slowest. The other bars (1ms to 6ms) are much lower, indicating faster execution times.

Code | C++

/*
 * Definition for a binary tree node.
 * struct TreeNode {
 * int val;
 * struct TreeNode *left;
 * struct TreeNode *right;
 * };
 */

class Solution {
public:
 vector<vector<int>> levelOrder(TreeNode* root) {
 vector<vector<int>> res;
 if(!root) return res;
 queue<TreeNode*> q;
 q.push(root);

 while(!q.empty()){
 int n=q.size();
 vector<int> ans;
 while(n--){
 TreeNode* temp=q.front();
 q.pop();
 ans.push_back(temp->val);
 if(temp->left) q.push(temp->left);
 if(temp->right) q.push(temp->right);
 }
 res.push_back(ans);
 }
 return res;
 }
};

Testcase Test Result

Case 1 Case 2 Case 3

root =
13 9 20 null null 15 71

Source

23°C Mostly clear

Search

ENG IN 19:22 10-03-2025

CU-Assignments/assignment5...Binary Tree Inorder Traversal - Binary Tree Inorder Traversal - BST Validation C++ Solution

leetcode.com/problems/binary-tree-inorder-traversal/submissions/1569147956/

Problem ListRunSubmit

DescriptionAcceptedEditorialSolutionsSubmissions

All Submissions

Accepted71 / 71 testcases passed

MohitBhal submitted at Mar 10, 2025 19:21

EditorialSolution

Runtime0 msBeats 100.00%

Memory10.76 MBBeats 88.32%

Analyze Complexity

100%

50%

0%

1ms2ms3ms4ms

1ms2ms3ms4ms

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) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
 * };
 */
class Solution {
public:
    void inorder(TreeNode* root, vector<int>& result) {
        if (!root) return;
        inorder(root->left, result);
        result.push_back(root->val);
    }
}
```

Code

C++Auto

```
1 /**
2  * Definition for 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(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
10 * };
11 */
12 class Solution {
13 public:
14     void inorder(TreeNode* root, vector<int>& result) {
15         if (!root) return;
16         inorder(root->left, result);
17         result.push_back(root->val);
18     }
19 }
```

SavedLn 27, Col 5

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2Case 3Case 4

Input

23°C Mostly clear

Search

ENG IN19:21 10-03-2025

CU-Assignments/assignment5...Lowest Common Ancestor of aBST Validation C++ Solution

leetcode.com/problems/lowest-common-ancestor-of-a-binary-tree/submissions/1569145079/

Problem ListRunSubmit

DescriptionAcceptedEditorialSolutionsSubmissions

All Submissions

Accepted32 / 32 testcases passedMohitBehal submitted at Mar 10, 2025 19:18

Runtime13 msBeats 46.27%

Memory17.36 MBBeats 67.78%

Runtime

0%10%20%

11ms21ms31ms41ms51ms

Code | C++

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q) {
        if (!root || 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;
    }
};
```

Code

C++Auto

```
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     TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q) {
14         if (!root || root == p || root == q) return root;
15
16         TreeNode* left = lowestCommonAncestor(root->left, p, q);
17         TreeNode* right = lowestCommonAncestor(root->right, p, q);
18
19         if (left && right) return root;
20         return left ? left : right;
21     }
22 }
```

SavedLn 19, Col 36

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2Case 3

Input

23°C Mostly clear

Search

ENG IN

19:18 10-03-2025

CU-Assignments/assignment5... Binary Tree Zigzag Level Order BST Validation C++ Solution

leetcode.com/problems/binary-tree-zigzag-level-order-traversal/submissions/1569142577/

Problem List Run Submit

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 33 / 33 testcases passed

MohitBehal submitted at Mar 10, 2025 19:15

Editorial Solution

Runtime 0 ms Beats 100.00% Memory 15.35 MB Beats 6.47%

Analyze Complexity

100% 50% 0%

1ms 2ms 3ms 4ms

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) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
 * };
 */
```

Code

```
26 for (int i = 0; i < size; ++i) {
27     TreeNode* node = q.front();
28     q.pop();
29
30     if (leftToRight)
31         level.push_back(node->val);
32     else
33         level.push_front(node->val);
34
35     if (node->left) q.push(node->left);
36     if (node->right) q.push(node->right);
37 }
38
39 result.push_back(vector<int>(level.begin(), level.end()));
40 leftToRight = !leftToRight;
41 }
42
43 return result;
44 }
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

23°C Mostly clear

Search

ENG IN 19:15 10-03-2025

CU-Assignments/assignment5... Symmetric Tree - LeetCode BST Validation C++ Solution

leetcode.com/problems/symmetric-tree/submissions/1569141128/

Problem List Run Submit Premium

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 200 / 200 testcases passed

MohitBehal submitted at Mar 10, 2025 19:14

Editorial Solution

Runtime 0 ms Beats 100.00% Memory 18.34 MB Beats 84.57%

Analyze Complexity

100%

0%

1ms 2ms 3ms 4ms

Code | C++

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */
```

```
6 struct TreeNode *right;
7 struct TreeNode() : val(0), left(nullptr), right(nullptr) {}
8 struct TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
9 struct TreeNode(int x, struct TreeNode *left, struct TreeNode *right) : val(x), left(left), right(right) {}
10 };
11
12 class Solution {
13 public:
14     bool isMirror(struct TreeNode* t1, struct TreeNode* t2) {
15         if (!t1 && !t2) return true;
16         if (!t1 || !t2 || t1->val != t2->val) return false;
17         return isMirror(t1->left, t2->right) && isMirror(t1->right, t2->left);
18     }
19
20     bool isSymmetric(struct TreeNode* root) {
21         if (!root) return true;
22         return isMirror(root->left, root->right);
23     }
24 };
```

Saved Ln 23, Col 6

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

23°C Mostly clear 19:14 10-03-2025