New tab | Classroom-802/ap-experiment-7- | Maximum Depth of Binary Tree -

https://leetcode.com/problems/maximum-depth-of-binary-tree/submissions/1597118907/

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 39 / 39 testcases passed

HrishabhGupta7292 submitted at Apr 05, 2025 10:12

Editorial | Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

19.00 MB | Beats 75.82%

Code | C++

```cpp
class Solution {
public:
    int maxDepth(TreeNode* root) {
        if (root == nullptr) {
            return 0; // base case: empty tree has depth 0
        }

        // Recursively find the depth of left and right subtrees
```

View more

More challenges

- 110. Balanced Binary Tree
- 111. Minimum Depth of Binary Tree

Code

C++ | Auto

```cpp
class Solution {
public:
    int maxDepth(TreeNode* root) {
        if (root == nullptr) {
            return 0; // base case: empty tree has depth 0
        }

        // Recursively find the depth of left and right subtrees
        int leftDepth = maxDepth(root->left);
        int rightDepth = maxDepth(root->right);

        // Return the greater of the two depths plus one for the current node
        return 1 + max(leftDepth, rightDepth);
    }
};

```

Saved | Ln 16, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

root =

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

ENG IN 10:12 05-04-2025

New tab | Classroom-802/ap-experiment-7- | Validate Binary Search Tree - Leet

https://leetcode.com/problems/validate-binary-search-tree/submissions/1597119627/

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 86 / 86 testcases passed

HrishabhGupta7292 submitted at Apr 05, 2025 10:13

Editorial | Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

22.00 MB | Beats 48.45%

Code | C++

```cpp
class Solution {
public:
    bool isValidBST(TreeNode* root) {
        return validate(root, LONG_MIN, LONG_MAX); // use long to avoid int overf
    }

private:
    bool validate(TreeNode* node, long minVal, long maxVal) {
```

View more

More challenges

- 94. Binary Tree Inorder Traversal
- 501. Find Mode in Binary Search Tree

Code

C++ | Auto

```cpp
class Solution {
public:
    bool isValidBST(TreeNode* root) {
        return validate(root, LONG_MIN, LONG_MAX); // use long to avoid int overflow
    }

private:
    bool validate(TreeNode* node, long minVal, long maxVal) {
        if (node == nullptr) return true;

        if (node->val <= minVal || node->val >= maxVal) {
            return false; // violates BST property
        }

        // left subtree must be < node->val
        // right subtree must be > node->val
        return validate(node->left, minVal, node->val) &&
               validate(node->right, node->val, maxVal);
    }
};

```

Saved | Ln 21, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

root =

[2,1,3]

ENG IN 10:13 05-04-2025

New tab | Classroom-802/ap-experiment-7- | Symmetric Tree - LeetCode

https://leetcode.com/problems/symmetric-tree/submissions/1597120164/

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 200 / 200 testcases passed

HrishabhGupta7292 submitted at Apr 05, 2025 10:14

Editorial | Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

18.48 MB | Beats 60.62%

Code | C++

```cpp
class Solution {
public:
    bool isSymmetric(TreeNode* root) {
        if (root == nullptr) return true;
        return isMirror(root->left, root->right);
    }

private:
```

View more

More challenges

- 776. Split BST
- 1245. Tree Diameter

Code

C++ | Auto

```cpp
class Solution {
public:
    bool isSymmetric(TreeNode* root) {
        if (root == nullptr) return true;
        return isMirror(root->left, root->right);
    }

private:
    bool isMirror(TreeNode* t1, TreeNode* t2) {
        if (t1 == nullptr && t2 == nullptr) return true;
        if (t1 == nullptr || t2 == nullptr) return false;

        return (t1->val == t2->val) &&
               isMirror(t1->left, t2->right) &&
               isMirror(t1->right, t2->left);
    }
};

```

Saved | Ln 18, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

root =

[1,2,2,3,4,4,3]

ENG IN 10:14 05-04-2025

New tab | Classroom-802/ap-experiment-7- | Binary Tree Level Order Traversal -

https://leetcode.com/problems/binary-tree-level-order-traversal/submissions/1597120866/

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 35 / 35 testcases passed

HrishabhGupta7292 submitted at Apr 05, 2025 10:15

Editorial | Solution

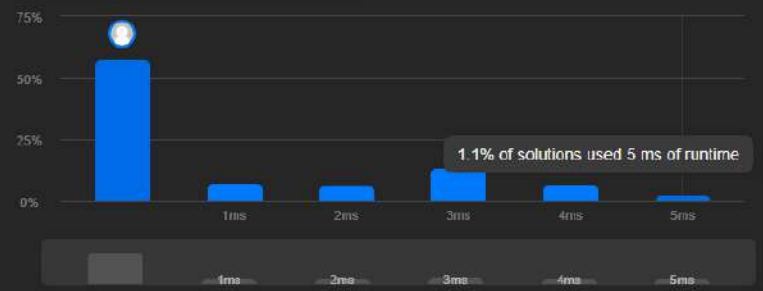
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

16.97 MB | Beats 88.36%



Code | C++

```cpp
class Solution {
public:
    vector<vector<int>> levelOrder(TreeNode* root) {
        vector<vector<int>> result;
        if (root == nullptr) return result;

        queue<TreeNode*> q;
        q.push(root);
```

View more

More challenges

- 103. Binary Tree Zigzag Level Order Traversal
- 107. Binary Tree Level Order Traversal II

Code

C++ | Auto

```cpp
vector<vector<int>> levelOrder(TreeNode* root) {
    vector<vector<int>> result;
    if (root == nullptr) return result;

    queue<TreeNode*> q;
    q.push(root);

    while (!q.empty()) {
        int levelSize = q.size(); // number of nodes at current level
        vector<int> level;

        for (int i = 0; i < levelSize; ++i) {
            TreeNode* current = q.front();
            q.pop();
            level.push_back(current->val);

            if (current->left) q.push(current->left);
            if (current->right) q.push(current->right);
        }

        result.push_back(level);
    }

    return result;
}
};
```

Saved | Ln 29, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

root =

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

ENG IN 10:15 05-04-2025

New tab | Classroom-802/ap-experiment-7- | Binary Tree Inorder Traversal - Lee

https://leetcode.com/problems/binary-tree-inorder-traversal/submissions/1597121828/

Problem List

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 71 / 71 testcases passed

HrishabhGupta7292 submitted at Apr 05, 2025 10:16

Editorial | Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

10.95 MB | Beats 35.81%

Code | C++

```cpp
class Solution {
public:
    vector<int> inorderTraversal(TreeNode* root) {
        vector<int> result;
        inorderHelper(root, result);
        return result;
    }
```

View more

More challenges

- 144. Binary Tree Preorder Traversal
- 145. Binary Tree Postorder Traversal

Run | Submit | Premium

Code | Submit Ctrl Enter

C++ | Auto

```cpp
class Solution {
public:
    vector<int> inorderTraversal(TreeNode* root) {
        vector<int> result;
        inorderHelper(root, result);
        return result;
    }

private:
    void inorderHelper(TreeNode* node, vector<int>& result) {
        if (node == nullptr) return;

        inorderHelper(node->left, result);     // visit left subtree
        result.push_back(node->val);           // visit node
        inorderHelper(node->right, result);    // visit right subtree
    }
};

```

Saved

Ln 18, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3 | Case 4

Input

root =

[1,null,2,3]

ENG IN 10:16 05-04-2025

New tab | Classroom-802/ap-experiment-7- | Binary Tree Inorder Traversal - Lee | Convert Sorted Array to Binary Se

https://leetcode.com/problems/convert-sorted-array-to-binary-search-tree/submissions/1597122623/

Problem List

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 31 / 31 testcases passed

HrishabhGupta7292 submitted at Apr 05, 2025 10:18

Editorial | Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

23.06 MB | Beats 23.05%

Code | C++

```cpp
class Solution {
public:
    TreeNode* sortedArrayToBST(vector<int>& nums) {
        return buildBST(nums, 0, nums.size() - 1);
    }

private:
    TreeNode* buildBST(vector<int>& nums, int left, int right) {
```

View more

More challenges

- 109. Convert Sorted List to Binary Search Tree

Run | Submit | Premium

Code

C++ | Auto

```cpp
class Solution {
public:
    TreeNode* sortedArrayToBST(vector<int>& nums) {
        return buildBST(nums, 0, nums.size() - 1);
    }

private:
    TreeNode* buildBST(vector<int>& nums, int left, int right) {
        if (left > right) return nullptr;

        int mid = left + (right - left) / 2;
        TreeNode* root = new TreeNode(nums[mid]);

        root->left = buildBST(nums, left, mid - 1);
        root->right = buildBST(nums, mid + 1, right);

        return root;
    }
};
```

Saved

Ln 20, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

- Case 1
- Case 2

Input

nums =

[-10,-3,0,5,9]

ENG IN

10:18

05-04-2025

New tab | Classroom-802/ap-experi | Page Not Found - LeetCo | Problems - LeetCode | Binary Zigzag Level Order | Binary Tree Zigzag Level

https://leetcode.com/problems/binary-tree-zigzag-level-order-traversal/submissions/1597124455/

Problem List

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 33 / 33 testcases passed

HrishabhGupta7292 submitted at Apr 05, 2025 10:20

Editorial | Solution

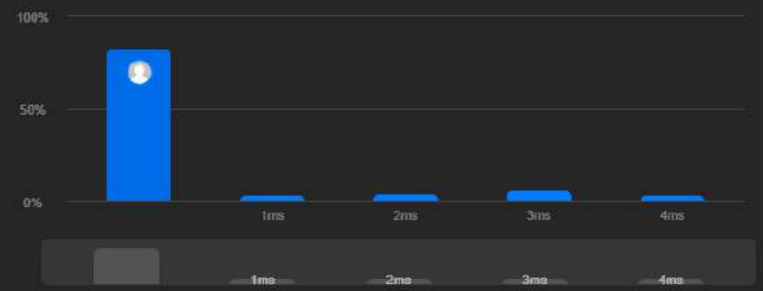
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

14.96 MB | Beats 93.76%



Code | C++

```cpp
class Solution {
public:
    vector<vector<int>> zigzagLevelOrder(TreeNode* root) {
        vector<vector<int>> result;
        if (!root) return result;

        queue<TreeNode*> q;
        q.push(root);
```

View more

More challenges

- 3417. Zigzag Grid Traversal With Skip

Run | Submit | Premium

Code

C++ | Auto

```cpp
class Solution {
public:
    vector<vector<int>> zigzagLevelOrder(TreeNode* root) {
        vector<vector<int>> result;
        if (!root) return result;

        queue<TreeNode*> q;
        q.push(root);
        bool leftToRight = true;

        while (!q.empty()) {
            int levelSize = q.size();
            vector<int> level(levelSize);  // Pre-size to allow index assignment

            for (int i = 0; i < levelSize; ++i) {
                TreeNode* node = q.front();
                q.pop();

                // Determine the index based on the direction
                int index = leftToRight ? i : (levelSize - 1 - i);
                level[index] = node->val;

                if (node->left) q.push(node->left);
                if (node->right) q.push(node->right);
            }

            result.push_back(level);
            leftToRight = !leftToRight;  // Flip direction
```

Saved

Ln 34, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

root =

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

ENG IN

10:21 05-04-2025

New tab | Classroom-802/ap-experiment-7 | Construct Binary Tree from Inorde | Kth Smallest Element in a BST - L | Populating Next Right Pointers in

https://leetcode.com/problems/construct-binary-tree-from-inorder-and-postorder-traversal/submissions/1597125135/

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 202 / 202 testcases passed

HrishabhGupta7292 submitted at Apr 05, 2025 10:22

Editorial | Solution

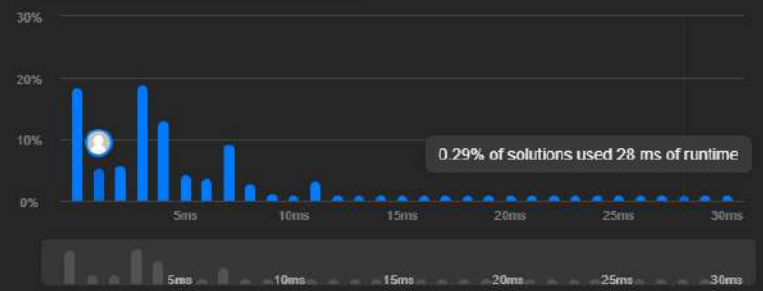
Runtime

1 ms | Beats 81.40%

Analyze Complexity

Memory

27.46 MB | Beats 59.60%



Code | C++

```
class Solution {
public:
    TreeNode* buildTree(vector<int>& inorder, vector<int>& postorder) {
        unordered_map<int, int> inorderIndex;
        for (int i = 0; i < inorder.size(); ++i)
            inorderIndex[inorder[i]] = i;

        int postIndex = postorder.size() - 1;
```

View more

More challenges

- 105. Construct Binary Tree from Preorder and Inorder Traversal

Code

C++ | Auto

```
1  class Solution {
2  public:
3      TreeNode* buildTree(vector<int>& inorder, vector<int>& postorder) {
4          unordered_map<int, int> inorderIndex;
5          for (int i = 0; i < inorder.size(); ++i)
6              inorderIndex[inorder[i]] = i;
7
8          int postIndex = postorder.size() - 1;
9          return build(inorder, postorder, inorderIndex, 0, inorder.size() - 1, postIndex);
10     }
11
12 private:
13     TreeNode* build(const vector<int>& inorder, const vector<int>& postorder,
14                     unordered_map<int, int>& inorderIndex,
15                     int inStart, int inEnd, int& postIndex) {
16         if (inStart > inEnd) return nullptr;
17
18         int rootVal = postorder[postIndex--];
19         TreeNode* root = new TreeNode(rootVal);
20
21         int inRoot = inorderIndex[rootVal];
22
23         // Important: build right subtree first (because we go backwards in postorder)
24         root->right = build(inorder, postorder, inorderIndex, inRoot + 1, inEnd, postIndex);
25         root->left = build(inorder, postorder, inorderIndex, inStart, inRoot - 1, postIndex);
26
27         return root;
28     }
```

Saved

Ln 30, Col 1

Testcase | Test Result

Case 1 | Case 2

inorder =

[9,3,15,20,7]

postorder =

Source

ENG IN 10:22 05-04-2025

New tab | Classroom-802/ap-experiment-7- | Kth Smallest Element in a BST - Le | Populating Next Right Pointers in

https://leetcode.com/problems/kth-smallest-element-in-a-bst/submissions/1597125704/

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 93 / 93 testcases passed

HrishabhGupta7292 submitted at Apr 05, 2025 10:22

Editorial | Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

24.22 MB | Beats 90.22%

Analyze Complexity

Code | C++

```cpp
class Solution {
public:
    int kthSmallest(TreeNode* root, int k) {
        int count = 0;
        int result = -1;
        inorder(root, k, count, result);
        return result;
    }
```

View more

More challenges

- 671. Second Minimum Node In a Binary Tree

Code

C++ | Auto

```cpp
class Solution {
public:
    int kthSmallest(TreeNode* root, int k) {
        int count = 0;
        int result = -1;
        inorder(root, k, count, result);
        return result;
    }

private:
    void inorder(TreeNode* node, int k, int& count, int& result) {
        if (!node) return;

        inorder(node->left, k, count, result);

        count++;
        if (count == k) {
            result = node->val;
            return;
        }

        inorder(node->right, k, count, result);
    }
};
```

Saved | Ln 25, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

root =

[3,1,4,null,2]

ENG IN 10:23 05-04-2025

New tab | Classroom-802/ap-experiment-7- | Populating Next Right Pointers in

https://leetcode.com/problems/populating-next-right-pointers-in-each-node/submissions/1597126461/

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 59 / 59 testcases passed

HrishabhGupta7292 submitted at Apr 05, 2025 10:24

Editorial | Solution

Runtime

13 ms | Beats 59.03%

Analyze Complexity

Memory

18.79 MB | Beats 98.41%

Code | C++

```cpp
class Solution {
public:
    Node* connect(Node* root) {
        if (!root) return nullptr;

        Node* level_start = root;

        while (level_start->left) {
```

View more

More challenges

- 117. Populating Next Right Pointers in Each Node II
- 199. Binary Tree Right Side View

Code

C++ | Auto

```cpp
class Solution {
public:
    Node* connect(Node* root) {
        if (!root) return nullptr;

        Node* level_start = root;

        while (level_start->left) {
            Node* curr = level_start;

            while (curr) {

                curr->left->next = curr->right;


                if (curr->next) {
                    curr->right->next = curr->next->left;
                }

                curr = curr->next;
            }


            level_start = level_start->left;
        }

        return root;
    }
```

Saved

Ln 23, Col 9

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

root =

[1,2,3,4,5,6,7]

ENG IN 10:24 05-04-2025