Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 39 / 39 testcases passed

divyanshu919891 submitted at Mar 10, 2025 19:46

Editorial | Solution

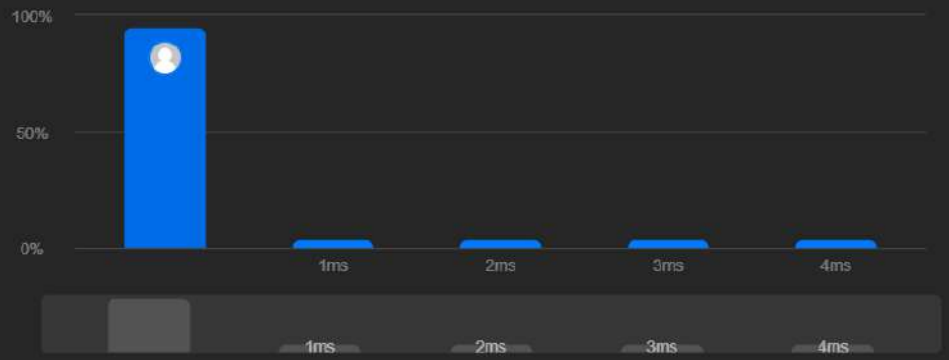
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

19.16 MB | Beats 13.19%



Code | C++

```cpp
class Solution {
public:
  int maxDepth(TreeNode* root) {
    if (root == nullptr)
```

Code

C++ | Auto

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

Saved

Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

Description | Accepted × | Editorial | Solutions | Submissions

← All Submissions

Accepted 86 / 86 testcases passed

divyanshu919891 submitted at Mar 10, 2025 19:47

Editorial | Solution

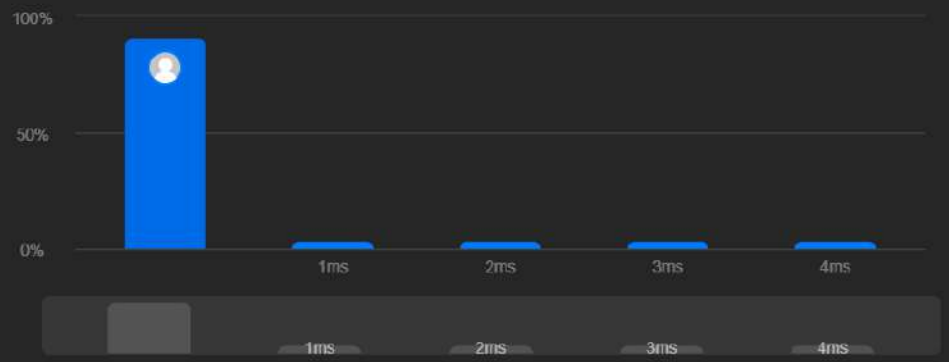
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

21.92 MB | Beats 48.51%



Code | C++

```cpp
class Solution {
 public:
  bool isValidBST(TreeNode* root) {
    return isValidBST(root, nullptr, nullptr);
```

Code

C++ | Auto

```cpp
class Solution {
 public:
  bool isValidBST(TreeNode* root) {
    return isValidBST(root, nullptr, nullptr);
  }

 private:
  bool isValidBST(TreeNode* root, TreeNode* minNode, TreeNode* maxNode) {
    if (root == nullptr)
      return true;
    if (minNode && root->val <= minNode->val)
      return false;
    if (maxNode && root->val >= maxNode->val)
      return false;

    return isValidBST(root->left, minNode, root) &&
           isValidBST(root->right, root, maxNode);
  }
};
```

Saved

Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

Description | Accepted × | Editorial | Solutions | Submissions

← All Submissions

Accepted 200 / 200 testcases passed

divyanshu919891 submitted at Mar 10, 2025 19:48

Editorial | Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

18.41 MB | Beats 59.30%

Code | C++

```cpp
class Solution {
 public:
  bool isSymmetric(TreeNode* root) {
    return isSymmetric(root, root);
```

Code

C++ | Auto

```cpp
class Solution {
public:
  bool isSymmetric(TreeNode* root) {
    return isSymmetric(root, root);
  }

private:
  bool isSymmetric(TreeNode* p, TreeNode* q) {
    if (!p || !q)
      return p == q;

    return p->val == q->val &&               //
           isSymmetric(p->left, q->right) &&  //
           isSymmetric(p->right, q->left);
  }
};
```

Saved

Ln 16, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

Description | Accepted | Editorial | Solutions | Submissions

← All Submissions

Accepted 33 / 33 testcases passed

divyanshu919891 submitted at Mar 10, 2025 19:49

Editorial | Solution

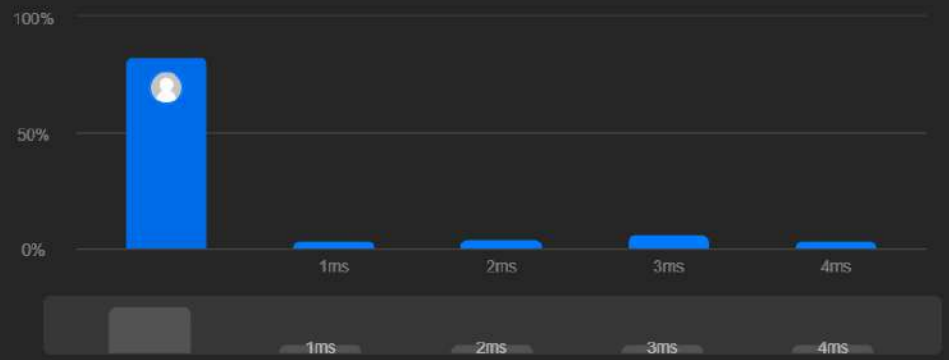
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

15.40 MB | Beats 6.47%



Code | C++

```cpp
class Solution {
 public:
  vector<vector<int>> zigzagLevelOrder(TreeNode* root) {
    if (root == nullptr)
```

Code

C++ | Auto

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

    vector<vector<int>> ans;
    deque<TreeNode*> dq{{root}};
    bool isLeftToRight = true;

    while (!dq.empty()) {
      vector<int> currLevel;
      for (int sz = dq.size(); sz > 0; --sz)
        if (isLeftToRight) {
          TreeNode* node = dq.front();
          dq.pop_front();
          currLevel.push_back(node->val);
          if (node->left)
            dq.push_back(node->left);
```

Saved

Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 32 / 32 testcases passed

divyanshu919891 submitted at Mar 10, 2025 19:50

Editorial | Solution

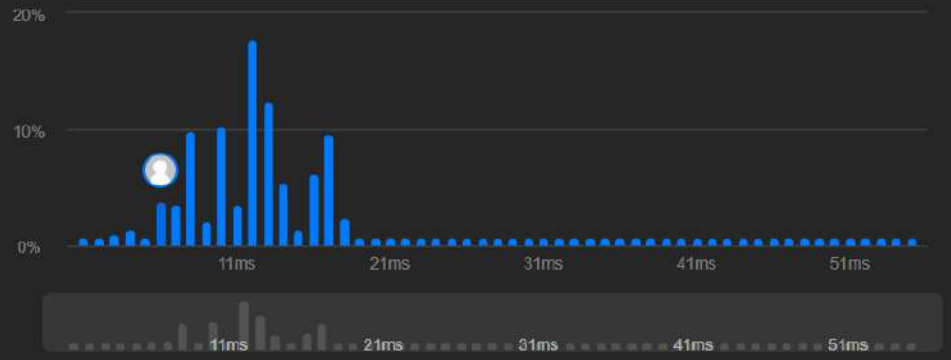
Runtime

6 ms | Beats 97.15%

Analyze Complexity

Memory

17.47 MB | Beats 41.55%



Code | C++

```cpp
class Solution {
public:
  TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q) {
    if (root == nullptr || root == p || root == q)
```

Code

C++ | Auto

```cpp
class Solution {
public:
  TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q) {
    if (root == nullptr || root == p || root == q)
      return root;
    TreeNode* left = lowestCommonAncestor(root->left, p, q);
    TreeNode* right = lowestCommonAncestor(root->right, p, q);
    if (left != nullptr && right != nullptr)
      return root;
    return left == nullptr ? right : left;
  }
};
```

Saved

Ln 12, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

Description | Accepted × | Editorial | Solutions | Submissions

← All Submissions

Accepted 71 / 71 testcases passed

divyanshu919891 submitted at Mar 10, 2025 19:51

Editorial | Solution

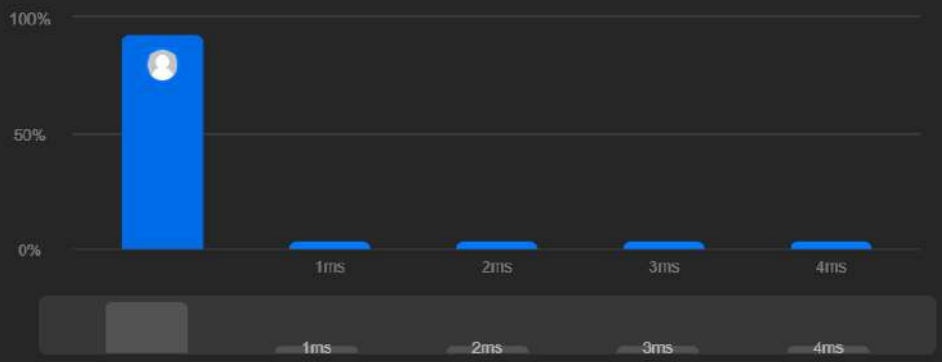
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

10.72 MB | Beats 88.32%



Code | C++

```cpp
class Solution {
 public:
  vector<int> inorderTraversal(TreeNode* root) {
    vector<int> ans;
```

Code

C++ | Auto

```cpp
class Solution {
public:
 vector<int> inorderTraversal(TreeNode* root) {
   vector<int> ans;
   stack<TreeNode*> stack;

   while (root != nullptr || !stack.empty()) {
     while (root != nullptr) {
       stack.push(root);
       root = root->left;
     }
     root = stack.top(), stack.pop();
     ans.push_back(root->val);
     root = root->right;
   }

   return ans;
 }
};
```

Saved

Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3 | Case 4

Input

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 35 / 35 testcases passed

divyanshu919891 submitted at Mar 10, 2025 19:52

Editorial | Solution

Runtime

3 ms | Beats 28.08%

Analyze Complexity

Memory

17.17 MB | Beats 44.34%

Code | C++

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

Code

C++ | Auto

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

  vector<vector<int>> ans;
  queue<TreeNode*> q{{root}};

  while (!q.empty()) {
   vector<int> currLevel;
   for (int sz = q.size(); sz > 0; --sz) {
    TreeNode* node = q.front();
    q.pop();
    currLevel.push_back(node->val);
    if (node->left)
     q.push(node->left);
    if (node->right)
     q.push(node->right);
```

Saved

Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

Description | Accepted × | Editorial | Solutions | Submissions

← All Submissions

Accepted 93 / 93 testcases passed

divyanshu919891 submitted at Mar 10, 2025 19:52

Editorial | Solution

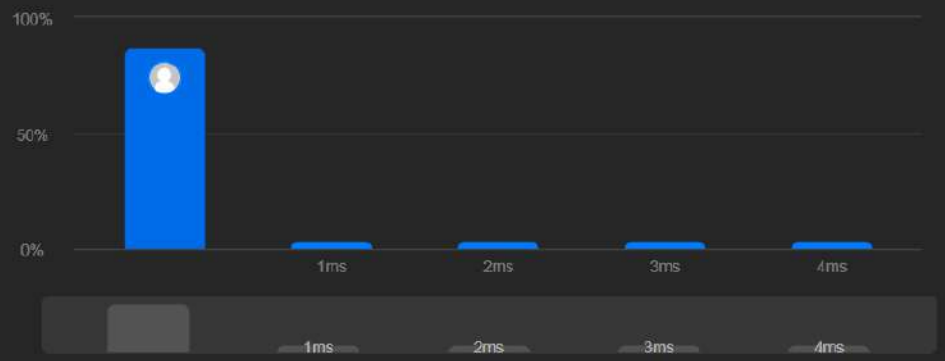
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

24.36 MB | Beats 67.94%



Code | C++

```cpp
class Solution {
 public:
  int kthSmallest(TreeNode* root, int k) {
    const int leftCount = countNodes(root->left);
```

Code

C++ | Auto

```cpp
public:
 int kthSmallest(TreeNode* root, int k) {
   const int leftCount = countNodes(root->left);

   if (leftCount == k - 1)
     return root->val;
   if (leftCount >= k)
     return kthSmallest(root->left, k);
   return kthSmallest(root->right, k - 1 - leftCount);  // leftCount < k
 }

private:
 int countNodes(TreeNode* root) {
   if (root == nullptr)
     return 0;
   return 1 + countNodes(root->left) + countNodes(root->right);
 }
};
```

Saved

Ln 19, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

Description | Accepted × | Editorial | Solutions | Submissions

← All Submissions

Accepted 59 / 59 testcases passed

divyanshu919891 submitted at Mar 10, 2025 19:53

Editorial | Solution

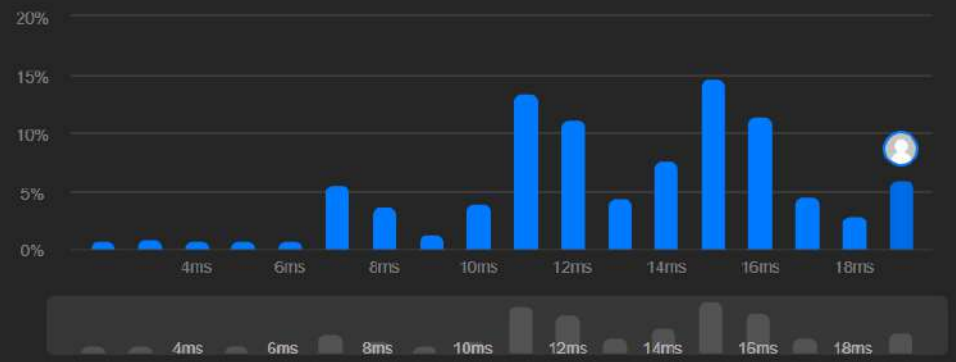
Runtime

22 ms | Beats 7.32%

Analyze Complexity

Memory

18.88 MB | Beats 90.91%



Code | C++

```cpp
class Solution {
 public:
  Node* connect(Node* root) {
    if (root == nullptr)
```

Code

C++ | Auto

```cpp
class Solution {
 public:
  Node* connect(Node* root) {
    if (root == nullptr)
      return nullptr;
    connectTwoNodes(root->left, root->right);
    return root;
  }

 private:
  void connectTwoNodes(Node* p, Node* q) {
    if (p == nullptr)
      return;
    p->next = q;
    connectTwoNodes(p->left, p->right);
    connectTwoNodes(q->left, q->right);
    connectTwoNodes(p->right, q->left);
  }
};
```

Saved

Ln 19, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 100 / 100 testcases passed

divyanshu919891 submitted at Mar 10, 2025 19:54

Editorial | Solution

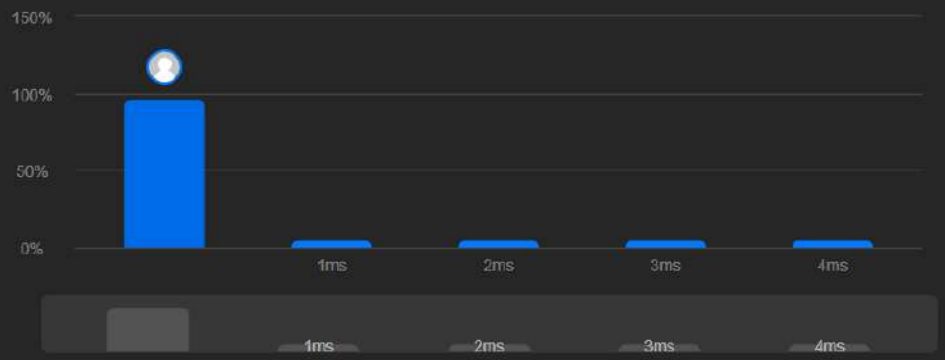
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

16.24 MB | Beats 23.61%



Code | C++

```cpp
class Solution {
 public:
  int sumOfLeftLeaves(TreeNode* root) {
    if (root == nullptr)
```

Code

C++ | Auto

```cpp
class Solution {
public:
  int sumOfLeftLeaves(TreeNode* root) {
    if (root == nullptr)
      return 0;

    int ans = 0;

    if (root->left) {
      if (root->left->left == nullptr && root->left->right == nullptr)
        ans += root->left->val;
      else
        ans += sumOfLeftLeaves(root->left);
    }
    ans += sumOfLeftLeaves(root->right);

    return ans;
  }
};
```

Saved

Ln 19, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input