

LINNNCODE

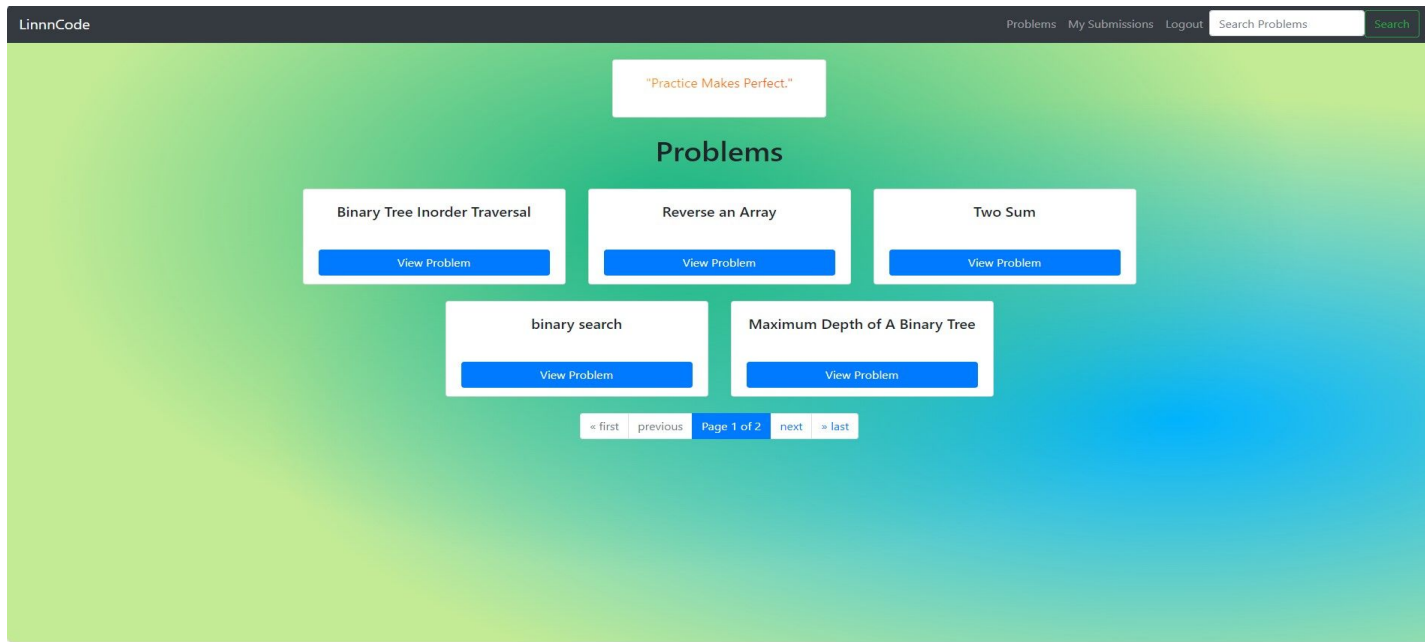


PRACTICE CODING AND PROBLEM-SOLVING ONLINE

By Yu-Heng Lin, and Linfeng Zhang
Demo: [LinnnCode](#)

OUR SOLUTION TO STUDENT'S LACK OF PRACTICE

- Students can actually code online!
- With diverse problems testing your coding skills



CODE!

- Solve the problem!

LinnCode

ProblemsMy SubmissionsLogoutSearch ProblemsSearch

Binary Tree Inorder Traversal

You are given a binary tree data structure, and you need to perform an inorder traversal on this tree. In inorder traversal, you first traverse the left subtree, then visit the root node, and finally traverse the right subtree. However, instead of simply visiting the nodes, you need to store the values of the visited nodes in a vector in the order they are visited.

The order of traversal of nodes is 4 → 2 → 5 → 1 → 3 → 6.

C++

```
1 // Structure of a Binary Tree Node
2 // struct Node {
3 //   int val;
4 //   shared_ptr<Node> left;
5 //   shared_ptr<Node> right;
6 //   Node(int val)
7 //   {
8 //     this->val = val;
9 //     this->left = nullptr;
10 //     this->right = nullptr;
11 //   }
12 // }
```

CODE EXECUTION

- Run the code and see if you pass

test2 ---> FAIL

test3 ---> FAIL

C++

```
1 // Structure of a Binary Tree Node
2 // struct Node {
3 //   int val;
4 //   shared_ptr<Node> left;
5 //   shared_ptr<Node> right;
6 //   Node(int val)
7 //   {
8 //     this->val = val;
9 //     this->left = nullptr;
10 //    this->right = nullptr;
11 //  }
12 // };
13
14
15
16 // Implement this
17 void inorder(shared_ptr<Node> node, vector<int>& res)
18 {
19 }
```

SUBMISSIONS

- If you failed, don't worry, see how others solve the problem!

LinnCode

ProblemsMy SubmissionsLogoutSearch ProblemsSearch

Submissions to Maximum Depth of A Binary Tree

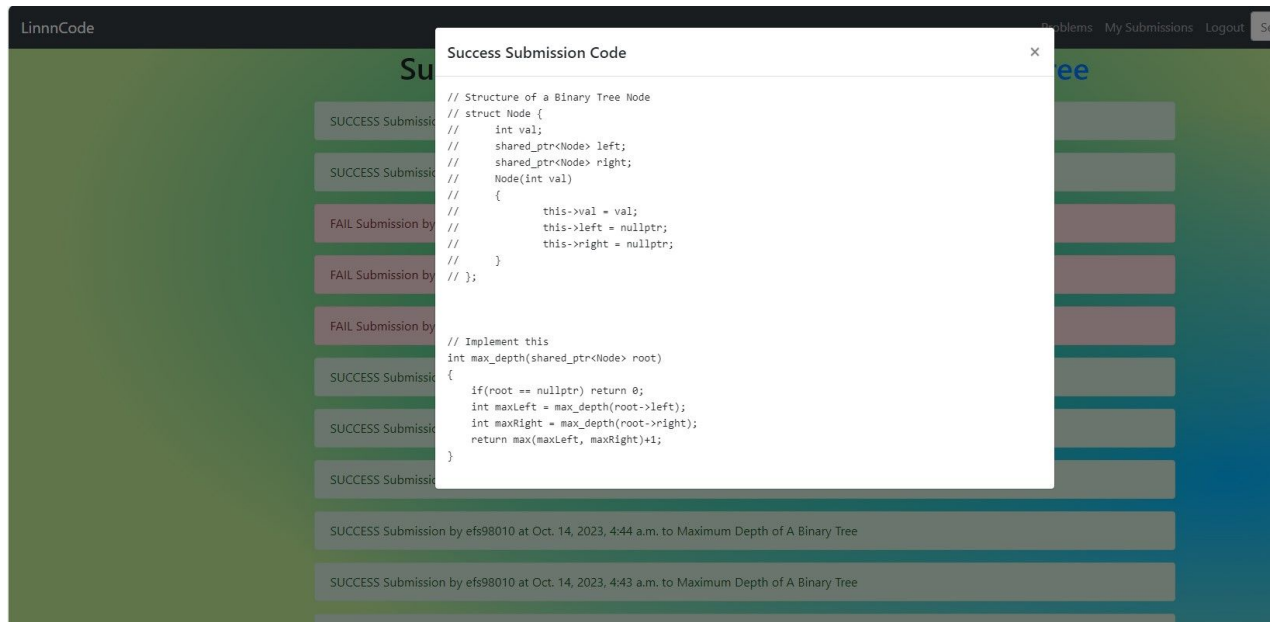
SUCCESS Submission by test1 at Oct. 14, 2023, 5:15 a.m. to Maximum Depth of A Binary Tree
SUCCESS Submission by efs98010 at Oct. 14, 2023, 4:48 a.m. to Maximum Depth of A Binary Tree
FAIL Submission by efs98010 at Oct. 14, 2023, 4:48 a.m. to Maximum Depth of A Binary Tree
FAIL Submission by efs98010 at Oct. 14, 2023, 4:47 a.m. to Maximum Depth of A Binary Tree
FAIL Submission by efs98010 at Oct. 14, 2023, 4:46 a.m. to Maximum Depth of A Binary Tree
SUCCESS Submission by efs98010 at Oct. 14, 2023, 4:46 a.m. to Maximum Depth of A Binary Tree
SUCCESS Submission by efs98010 at Oct. 14, 2023, 4:45 a.m. to Maximum Depth of A Binary Tree
SUCCESS Submission by efs98010 at Oct. 14, 2023, 4:44 a.m. to Maximum Depth of A Binary Tree
SUCCESS Submission by efs98010 at Oct. 14, 2023, 4:44 a.m. to Maximum Depth of A Binary Tree
SUCCESS Submission by efs98010 at Oct. 14, 2023, 4:43 a.m. to Maximum Depth of A Binary Tree
SUCCESS Submission by efs98010 at Oct. 14, 2023, 4:43 a.m. to Maximum Depth of A Binary Tree

« first previous Page 1 of 2 next > last

VIEW OTHERS/YOUR CODE

- Learn from your own mistakes

Or others' mistakes!



The screenshot shows a web application interface for a coding platform. At the top, there is a navigation bar with the text "LinnCode" on the left and "Problems My Submissions Logout" on the right. Below the navigation bar, there is a list of submissions. Each submission entry includes a status (e.g., "SUCCESS Submission", "FAIL Submission by"), a username, and a timestamp. A modal window titled "Success Submission Code" is open, displaying the C++ code for a binary tree node structure and a function to calculate the maximum depth of a binary tree.

```
// Structure of a Binary Tree Node
// struct Node {
//     int val;
//     shared_ptr<Node> left;
//     shared_ptr<Node> right;
//     Node(int val)
//     {
//         this->val = val;
//         this->left = nullptr;
//         this->right = nullptr;
//     }
// };

// Implement this
int max_depth(shared_ptr<Node> root)
{
    if(root == nullptr) return 0;
    int maxLeft = max_depth(root->left);
    int maxRight = max_depth(root->right);
    return max(maxLeft, maxRight)+1;
}
```

Below the modal window, the submission list continues with entries like "SUCCESS Submission by efs98010 at Oct. 14, 2023, 4:44 a.m. to Maximum Depth of A Binary Tree".

THANK YOU!