

Prompt	Scratchpad	Our Solution(s)	Video Explanation	Run Code
--------	------------	-----------------	-------------------	----------

Solution 1	Solution 2
------------	------------

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using namespace std;
4
5 ▼ class BinaryTree {
6     public:
7         int value;
8         BinaryTree *left;
9         BinaryTree *right;
10
11 ▼ BinaryTree(int value) {
12     this->value = value;
13     left = NULL;
14     right = NULL;
15 }
16 };
17
18 // Average case: when the tree is balanced
19 // O(n) time | O(h) space - where n is the number of nodes in
20 // the Binary Tree and h is the height of the Binary Tree
21 ▼ int nodeDepths(BinaryTree *root, int depth = 0) {
22     if (root == NULL)
23         return 0;
24     return depth + nodeDepths(root->left, depth + 1) +
25             nodeDepths(root->right, depth + 1);
26 }
```

