

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

Solution 1	Solution 2	Solution 3	Solution 4
<pre>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 int nodeDepths(BinaryTree *node, int depth = 0); 19 20 // Average case: when the tree is balanced 21 // O(nlog(n)) time O(h) space - where n is the number of nodes in 22 // the Binary Tree and h is the height of the Binary Tree 23 ▼ int allKindsOfNodeDepths(BinaryTree *root) { 24 if (root == NULL) 25 return 0; 26 return allKindsOfNodeDepths(root->left) + allKindsOfNodeDepths(root->right) + 27 nodeDepths(root); 28 } 29 30 ▼ int nodeDepths(BinaryTree *node, int depth) { 31 if (node == NULL) 32 return 0; 33 return depth + nodeDepths(node->left, depth + 1) + 34 nodeDepths(node->right, depth + 1); 35 }</pre>			

