

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-&gt;value = value; 13     left = NULL; 14     right = NULL; 15 } 16 }; 17 18 ▼ struct TreeInfo { 19     int numNodesInTree; 20     int sumOfDepths; 21     int sumOfAllDepths; 22 }; 23 24 TreeInfo getTreeInfo(BinaryTree *tree); 25 26 // Average case: when the tree is balanced 27 // O(n) time   O(h) space - where n is the number of nodes in 28 // the Binary Tree and h is the height of the Binary Tree 29 ▼ int allKindsOfNodeDepths(BinaryTree *root) { 30     return getTreeInfo(root).sumOfAllDepths; 31 } 32 33 ▼ TreeInfo getTreeInfo(BinaryTree *tree) { 34     ▼ if (tree == NULL) { 35         return TreeInfo{0, 0, 0}; 36     } 37 38     TreeInfo leftTreeInfo = getTreeInfo(tree-&gt;left); 39     TreeInfo rightTreeInfo = getTreeInfo(tree-&gt;right); 40 41     int sumOfLeftDepths = leftTreeInfo.sumOfDepths + leftTreeInfo.numNodesInTree; 42     int sumOfRightDepths = 43         rightTreeInfo.sumOfDepths + rightTreeInfo.numNodesInTree; 44 45     int numNodesInTree = 46         1 + leftTreeInfo.numNodesInTree + rightTreeInfo.numNodesInTree; 47     int sumOfDepths = sumOfLeftDepths + sumOfRightDepths; 48     int sumOfAllDepths = 49         sumOfDepths + leftTreeInfo.sumOfAllDepths + rightTreeInfo.sumOfAllDepths; 50 51 ▼ return TreeInfo{ 52     numNodesInTree, 53     sumOfDepths, 54     sumOfAllDepths, 55 }; 56 }</pre>			

