

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 ▾ class Program { 4 ▾   class BinaryTree { 5       var value: Int 6       var left: BinaryTree? 7       var right: BinaryTree? 8 9 ▾     init(value: Int) { 10         self.value = value 11     } 12 } 13 14 // Average case: when the tree is balanced 15 // O(nlog(n)) time   O(h) space - where n is the number of nodes in 16 // the Binary Tree and h is the height of the Binary Tree 17 ▾ static func allKindsOfNodeDepths(_ root: BinaryTree?) -&gt; Int { 18 ▾   if let node = root { 19       return allKindsOfNodeDepths(node.left) + allKindsOfNodeDepths(node.right) + nodeDepths(root, 0) 20   } 21   return 0 22 } 23 24 ▾ static func nodeDepths(_ root: BinaryTree?, _ depth: Int = 0) -&gt; Int { 25 ▾   if let tree = root { 26       return depth + nodeDepths(tree.left, depth + 1) + nodeDepths(tree.right, depth + 1) 27   } 28   return 0 29 } 30 } 31</pre>			

