

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 package main 4 5 ▾ type BinaryTree struct { 6 Value int 7 Left, Right *BinaryTree 8 } 9 10 // Average case: when the tree is balanced 11 // O(nlog(n)) time O(h) space - where n is the number of nodes in 12 // the Binary Tree and h is the height of the Binary Tree 13 ▾ func AllKindsOfNodeDepths(root *BinaryTree) int { 14 ▾ if root == nil { 15 return 0 16 } 17 return AllKindsOfNodeDepths(root.Left) + AllKindsOfNodeDepths(root.Right) + nodeDepths(root, 0) 18 } 19 20 ▾ func nodeDepths(node *BinaryTree, depth int) int { 21 ▾ if node == nil { 22 return 0 23 } 24 return depth + nodeDepths(node.Left, depth+1) + nodeDepths(node.Right, depth+1) 25 } 26</pre>			

