

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 ▼ type TreeInfo struct { 11 NumNodesInTree int 12 SumOfDepths int 13 SumOfAllDepths int 14 } 15 16 // Average case: when the tree is balanced 17 // O(n) time O(h) space - where n is the number of nodes in 18 // the Binary Tree and h is the height of the Binary Tree 19 ▼ func AllKindsOfNodeDepths(root *BinaryTree) int { 20 return getTreeInfo(root).SumOfAllDepths 21 } 22 23 ▼ func getTreeInfo(tree *BinaryTree) TreeInfo { 24 ▼ if tree == nil { 25 return TreeInfo{} 26 } 27 28 leftInfo, rightInfo := getTreeInfo(tree.Left), getTreeInfo(tree.Right) 29 30 sumOfLeftDepths := leftInfo.SumOfDepths + leftInfo.NumNodesInTree 31 sumOfRightDepths := rightInfo.SumOfDepths + rightInfo.NumNodesInTree 32 33 numNodesInTree := 1 + leftInfo.NumNodesInTree + rightInfo.NumNodesInTree 34 sumOfDepths := sumOfLeftDepths + sumOfRightDepths 35 sumOfAllDepths := sumOfDepths + leftInfo.SumOfAllDepths + rightInfo.SumOfAllDepths 36 37 return TreeInfo{NumNodesInTree: numNodesInTree, SumOfDepths: sumOfDepths, SumOfAllDepths: sumOfAllDepths} 38 } 39</pre>			

