

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 ▼ import java.util.*; 4 5 ▼ class Program { 6     // Average case: when the tree is balanced 7     // O(nlog(n)) time   O(h) space - where n is the number of nodes in 8     // the Binary Tree and h is the height of the Binary Tree 9 ▼ public static int allKindsOfNodeDepths(BinaryTree root) { 10     int sumOfAllDepths = 0; 11     List&lt;BinaryTree&gt; stack = new ArrayList&lt;BinaryTree&gt;(); 12     stack.add(root); 13 ▼ while (stack.size() &gt; 0) { 14         BinaryTree node = stack.remove(stack.size() - 1); 15         if (node == null) continue; 16         sumOfAllDepths += nodeDepths(node, 0); 17         stack.add(node.left); 18         stack.add(node.right); 19     } 20     return sumOfAllDepths; 21 } 22 23 ▼ public static int nodeDepths(BinaryTree node, int depth) { 24     if (node == null) return 0; 25     return depth + nodeDepths(node.left, depth + 1) + nodeDepths(node.right, depth + 1); 26 } 27 28 ▼ static class BinaryTree { 29     int value; 30     BinaryTree left; 31     BinaryTree right; 32 33 ▼ public BinaryTree(int value) { 34     this.value = value; 35     left = null; 36     right = null; 37 } 38 } 39 } 40</pre>			

