

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  # Average case: when the tree is balanced 4  # O(nlog(n)) time   O(h) space - where n is the number of nodes in 5  # the Binary Tree and h is the height of the Binary Tree 6  ▾ def allKindsOfNodeDepths(root): 7      sumOfAllDepths = 0 8      stack = [root] 9  ▾      while len(stack) &gt; 0: 10         node = stack.pop() 11  ▾         if node is None: 12             continue 13         sumOfAllDepths += nodeDepths(node) 14         stack.append(node.left) 15         stack.append(node.right) 16     return sumOfAllDepths 17 18 19  ▾ def nodeDepths(node, depth=0): 20  ▾      if node is None: 21         return 0 22         return depth + nodeDepths(node.left, depth + 1) + nodeDepths(node.right, depth + 1) 23 24 25  # This is the class of the input binary tree. 26  ▾ class BinaryTree: 27  ▾      def __init__(self, value): 28         self.value = value 29         self.left = None 30         self.right = None 31</pre>			

