

AlgoExpert

Quad Layout

Python

12px

Sublime

Monokai

Prompt

Scratchpad

Our Solution(s)

Video Explanation

Run Code

Solution 1

Solution 2

Solution 3

Solution 4

```
1  # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  # Average case: when the tree is balanced
4  # O(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      return getTreeInfo(root).sumOfAllDepths
8
9
10 ▾ def getTreeInfo(tree):
11 ▾     if tree is None:
12         return TreeInfo(0, 0, 0)
13
14     leftTreeInfo = getTreeInfo(tree.left)
15     rightTreeInfo = getTreeInfo(tree.right)
16
17     sumOfLeftDepths = leftTreeInfo.sumOfDepths + leftTreeInfo.numNodesInTree
18     sumOfRightDepths = rightTreeInfo.sumOfDepths + rightTreeInfo.numNodesInTree
19
20     numNodesInTree = 1 + leftTreeInfo.numNodesInTree + rightTreeInfo.numNodesInTree
21     sumOfDepths = sumOfLeftDepths + sumOfRightDepths
22     sumOfAllDepths = sumOfDepths + leftTreeInfo.sumOfAllDepths + rightTreeInfo.sumOfAllDepths
23
24     return TreeInfo(numNodesInTree, sumOfDepths, sumOfAllDepths)
25
26
27 ▾ class TreeInfo:
28 ▾     def __init__(self, numNodesInTree, sumOfDepths, sumOfAllDepths):
29         self.numNodesInTree = numNodesInTree
30         self.sumOfDepths = sumOfDepths
31         self.sumOfAllDepths = sumOfAllDepths
32
33
34     # This is the class of the input binary tree.
35 ▾ class BinaryTree:
36 ▾     def __init__(self, value):
37         self.value = value
38         self.left = None
39         self.right = None
40
```

