AlgoExpert Quad Layout Python 12px Sublime Monok

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# Average case: when the tree is balanced
# O(n) time | O(h) space - where n is the number of nodes in
# the Binary Tree and h is the height of the Binary Tree
def nodeDepths(root):
    sumOfDepths = 0
    stack = [{"node": root, "depth": 0}]
    while len(stack) > 0:
        nodeInfo = stack.pop()
        node, depth = nodeInfo["node"], nodeInfo["depth"]
        if node is None:
            continue
        sumOfDepths += depth
        stack.append({"node": node.left, "depth": depth + 1})
        stack.append({"node": node.right, "depth": depth + 1})
    return sumOfDepths


# This is the class of the input binary tree.
class BinaryTree:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None

```