

AlgoExpert

Quad Layout

C#

12px

Sublime

Monok

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 using System.Collections.Generic;
4
5 public class Program {
6     // Average case: when the tree is balanced
7     // O(n) time | O(n) space - where n is the number of nodes in the Binary Tree
8     public static int AllKindsOfNodeDepths(BinaryTree root) {
9         Dictionary<BinaryTree, int> nodeCounts = new Dictionary<BinaryTree, int>();
10        Dictionary<BinaryTree, int> nodeDepths = new Dictionary<BinaryTree, int>();
11        addNodeCounts(root, nodeCounts);
12        addNodeDepths(root, nodeDepths, nodeCounts);
13        return sumAllNodeDepths(root, nodeDepths);
14    }
15
16    public static int sumAllNodeDepths(BinaryTree node,
17    Dictionary<BinaryTree, int> nodeDepths) {
18        if (node == null) return 0;
19        return sumAllNodeDepths(node.left, nodeDepths) + sumAllNodeDepths(node.right,
20        nodeDepths) +
21        nodeDepths[node];
22    }
23
24    public static void addNodeDepths(BinaryTree node, Dictionary<BinaryTree, int> nodeDepths,
25    Dictionary<BinaryTree, int> nodeCounts) {
26        nodeDepths[node] = 0;
27        if (node.left != null) {
28            addNodeDepths(node.left, nodeDepths, nodeCounts);
29            nodeDepths[node] = nodeDepths[node] + nodeDepths[node.left] +
30            nodeCounts[node.left];
31        }
32        if (node.right != null) {
33            addNodeDepths(node.right, nodeDepths, nodeCounts);
34            nodeDepths[node] = nodeDepths[node] + nodeDepths[node.right] +
35            nodeCounts[node.right];
36        }
37    }
38
39    public static void addNodeCounts(BinaryTree node, Dictionary<BinaryTree, int> nodeCounts) {
40        nodeCounts[node] = 1;
41        if (node.left != null) {
42            addNodeCounts(node.left, nodeCounts);
43            nodeCounts[node] = nodeCounts[node] + nodeCounts[node.left];
44        }
45        if (node.right != null) {
46            addNodeCounts(node.right, nodeCounts);
47            nodeCounts[node] = nodeCounts[node] + nodeCounts[node.right];
48        }
49    }
50
51    public class BinaryTree {
52        public int value;
53        public BinaryTree left;
54        public BinaryTree right;
55
56        public BinaryTree(int value) {
57            this.value = value;
58            left = null;
59            right = null;
60        }
61    }
62 }
63
```

