

AlgoExpert

Quad LayoutGo12pxSublimeMonokai

Prompt	Scratchpad	Our Solution(s)	Video Explanation	Run Code
--------	------------	-----------------	-------------------	----------

Solution 1Solution 2Solution 3Solution 4

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 ▾ type BinaryTree struct {
6     Value      int
7     Left, Right *BinaryTree
8 }
9
10 // Average case: when the tree is balanced
11 // O(n) time | O(n) space - where n is the number of nodes in the Binary Tree
12 ▾ func AllKindsOfNodeDepths(root *BinaryTree) int {
13     nodeCounts, nodeDepths := map[*BinaryTree]int{}, map[*BinaryTree]int{}
14     addNodeCounts(root, nodeCounts)
15     addNodeDepths(root, nodeDepths, nodeCounts)
16     return sumAllNodeDepths(root, nodeDepths)
17 }
18
19 ▾ func sumAllNodeDepths(node *BinaryTree, nodeDepths map[*BinaryTree]int) int {
20     ▾ if node == nil {
21         return 0
22     }
23     return sumAllNodeDepths(node.Left, nodeDepths) + sumAllNodeDepths(node.Right, nodeDepths) + nodeDepths[node]
24 }
25
26 ▾ func addNodeDepths(node *BinaryTree, nodeDepths, nodeCounts map[*BinaryTree]int) {
27     nodeDepths[node] = 0
28     ▾ if node.Left != nil {
29         addNodeDepths(node.Left, nodeDepths, nodeCounts)
30         nodeDepths[node] = nodeDepths[node] + nodeDepths[node.Left] + nodeCounts[node.Left]
31     }
32     ▾ if node.Right != nil {
33         addNodeDepths(node.Right, nodeDepths, nodeCounts)
34         nodeDepths[node] = nodeDepths[node] + nodeDepths[node.Right] + nodeCounts[node.Right]
35     }
36 }
37
38 ▾ func addNodeCounts(node *BinaryTree, nodeCounts map[*BinaryTree]int) {
39     nodeCounts[node] = 1
40     ▾ if node.Left != nil {
41         addNodeCounts(node.Left, nodeCounts)
42         nodeCounts[node] = nodeCounts[node] + nodeCounts[node.Left]
43     }
44     ▾ if node.Right != nil {
45         addNodeCounts(node.Right, nodeCounts)
46         nodeCounts[node] = nodeCounts[node] + nodeCounts[node.Right]
47     }
48 }
49
```

