

AlgoExpert

Quad LayoutSwift12pxSublimeMonokai

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

Solution 1Solution 2Solution 3Solution 4

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 ▾ class Program {
4 ▾   class BinaryTree {
5       var value: Int
6       var left: BinaryTree?
7       var right: BinaryTree?
8
9 ▾   init(value: Int) {
10       self.value = value
11   }
12 }
13
14 // Average case: when the tree is balanced
15 // O(nlog(n)) time | O(h) space - where n is the number of nodes in
16 // the Binary Tree and h is the height of the Binary Tree
17 ▾ static func allKindsOfNodeDepths(_ root: BinaryTree?) -> Int {
18     var sumOfDepths = 0
19     var stack = [root]
20 ▾   while stack.count > 0 {
21       var node = stack[stack.count - 1]
22       stack.removeLast()
23 ▾     if node == nil {
24       continue
25     }
26
27     sumOfDepths += nodeDepths(node, 0)
28     stack.append(node?.left)
29     stack.append(node?.right)
30   }
31   return sumOfDepths
32 }
33
34 ▾ static func nodeDepths(_ root: BinaryTree?, _ depth: Int = 0) -> Int {
35 ▾   if let tree = root {
36     return depth + nodeDepths(tree.left, depth + 1) + nodeDepths(tree.right, depth + 1)
37   }
38   return 0
39 }
40 }
41
```

