

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

Solution 1Solution 2

```
1  // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  ▼ import java.util.*;
4
5  ▼ class Program {
6      // Average case: when the tree is balanced
7      // O(n) time | O(h) space - where n is the number of nodes in
8      // the Binary Tree and h is the height of the Binary Tree
9  ▼ public static int nodeDepths(BinaryTree root) {
10     return nodeDepthsHelper(root, 0);
11 }
12
13 ▼ public static int nodeDepthsHelper(BinaryTree root, int depth) {
14     if (root == null) return 0;
15     return depth + nodeDepthsHelper(root.left, depth + 1) + nodeDepthsHelper(root.right, depth + 1);
16 }
17
18 ▼ static class BinaryTree {
19     int value;
20     BinaryTree left;
21     BinaryTree right;
22
23 ▼ public BinaryTree(int value) {
24     this.value = value;
25     left = null;
26     right = null;
27 }
28 }
29 }
30
```

