```
1  // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3
4  public class Program {
5    // Average case: when the tree is balanced
6    // O(n) time | O(h) space - where n is the number of nodes in
7    // the Binary Tree and h is the height of the Binary Tree
8    public static int AllKindsOfNodeDepths(BinaryTree root) {
9      return getTreeInfo(root).sumOfAllDepths;
10   }
11
12   public static TreeInfo getTreeInfo(BinaryTree tree) {
13     if (tree == null) {
14       return new TreeInfo(0, 0, 0);
15     }
16
17     TreeInfo leftTreeInfo = getTreeInfo(tree.left);
18     TreeInfo rightTreeInfo = getTreeInfo(tree.right);
19
20     int sumOfLeftDepths = leftTreeInfo.sumOfDepths + leftTreeInfo.numNodesInTree;
21     int sumOfRightDepths =
22       rightTreeInfo.sumOfDepths + rightTreeInfo.numNodesInTree;
23
24     int numNodesInTree =
25       1 + leftTreeInfo.numNodesInTree + rightTreeInfo.numNodesInTree;
26     int sumOfDepths = sumOfLeftDepths + sumOfRightDepths;
27     int sumOfAllDepths =
28       sumOfDepths + leftTreeInfo.sumOfAllDepths + rightTreeInfo.sumOfAllDepths;
29
30     return new TreeInfo(numNodesInTree, sumOfDepths, sumOfAllDepths);
31   }
32
33   public class TreeInfo {
34     public int numNodesInTree;
35     public int sumOfDepths;
36     public int sumOfAllDepths;
37
38     public TreeInfo(int numNodesInTree, int sumOfDepths, int sumOfAllDepths) {
39       this.numNodesInTree = numNodesInTree;
40       this.sumOfDepths = sumOfDepths;
41       this.sumOfAllDepths = sumOfAllDepths;
42     }
43   }
44
45   public class BinaryTree {
46     public int value;
47     public BinaryTree left;
48     public BinaryTree right;
49
50     public BinaryTree(int value) {
51       this.value = value;
52       left = null;
53       right = null;
54     }
55   }
56 }
57
```