```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

import java.util.*;

class Program {
  // Average case: when the tree is balanced
  // O(n) time | O(h) space - where n is the number of nodes in
  // the Binary Tree and h is the height of the Binary Tree
  public static int allKindsOfNodeDepths(BinaryTree root) {
    return getTreeInfo(root).sumOfAllDepths;
  }

  public static TreeInfo getTreeInfo(BinaryTree tree) {
    if (tree == null) {
      return new TreeInfo(0, 0, 0);
    }

    TreeInfo leftTreeInfo = getTreeInfo(tree.left);
    TreeInfo rightTreeInfo = getTreeInfo(tree.right);

    int sumOfLeftDepths = leftTreeInfo.sumOfDepths + leftTreeInfo.numNodesInTree;
    int sumOfRightDepths =
        rightTreeInfo.sumOfDepths + rightTreeInfo.numNodesInTree;

    int numNodesInTree =
        1 + leftTreeInfo.numNodesInTree + rightTreeInfo.numNodesInTree;
    int sumOfDepths = sumOfLeftDepths + sumOfRightDepths;
    int sumOfAllDepths =
        sumOfDepths + leftTreeInfo.sumOfAllDepths + rightTreeInfo.sumOfAllDepths;

    return new TreeInfo(numNodesInTree, sumOfDepths, sumOfAllDepths);
  }

  static class TreeInfo {
    public int numNodesInTree;
    public int sumOfDepths;
    public int sumOfAllDepths;

    public TreeInfo(int numNodesInTree, int sumOfDepths, int sumOfAllDepths) {
      this.numNodesInTree = numNodesInTree;
      this.sumOfDepths = sumOfDepths;
      this.sumOfAllDepths = sumOfAllDepths;
    }
  }

  static class BinaryTree {
    int value;
    BinaryTree left;
    BinaryTree right;

    public BinaryTree(int value) {
      this.value = value;
      left = null;
      right = null;
    }
  }
}
```