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

package main

type BinaryTree struct {
  Value       int
  Left, Right *BinaryTree
}

type TreeInfo struct {
  NumNodesInTree int
  SumOfDepths    int
  SumOfAllDepths int
}

// 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
func AllKindsOfNodeDepths(root *BinaryTree) int {
  return getTreeInfo(root).SumOfAllDepths
}

func getTreeInfo(tree *BinaryTree) TreeInfo {
  if tree == nil {
    return TreeInfo{}
  }

  leftInfo, rightInfo := getTreeInfo(tree.Left), getTreeInfo(tree.Right)

  sumOfLeftDepths := leftInfo.SumOfDepths + leftInfo.NumNodesInTree
  sumOfRightDepths := rightInfo.SumOfDepths + rightInfo.NumNodesInTree

  numNodesInTree := 1 + leftInfo.NumNodesInTree + rightInfo.NumNodesInTree
  sumOfDepths := sumOfLeftDepths + sumOfRightDepths
  sumOfAllDepths := sumOfDepths + leftInfo.SumOfAllDepths + rightInfo.SumOfAllDepths

  return TreeInfo{NumNodesInTree: numNodesInTree, SumOfDepths: sumOfDepths, SumOfAllDepths: sumOfAllDepths}
}

```