AlgoExpert

Quad Layout | JavaScript | 12px | Sublime | Monok

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

Solution 1 | Solution 2 | Solution 3 | Solution 4

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

// Average case: when the tree is balanced
// O(nlog(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
function allKindsOfNodeDepths(root) {
  let sumOfAllDepths = 0;
  let stack = [root];
  while (stack.length > 0) {
    const node = stack.pop();
    if (node === null) continue;
    sumOfAllDepths += nodeDepths(node);
    stack.push(node.left);
    stack.push(node.right);
  }
  return sumOfAllDepths;
}

function nodeDepths(node, depth = 0) {
  if (node === null) return 0;
  return depth + nodeDepths(node.left, depth + 1) + nodeDepths(node.right, depth + 1);
}

// This is the class of the input binary tree.
class BinaryTree {
  constructor(value) {
    this.value = value;
    this.left = null;
    this.right = null;
  }
}

exports.allKindsOfNodeDepths = allKindsOfNodeDepths;
```