

Prompt

Scratchpad

Our Solution(s)

Video Explanation

**Difficulty:** ■ **Category:** Binary Trees **Successful Submissions:** 41,949+

## Node Depths

The distance between a node in a Binary Tree and the tree's root is called the node's depth.

Write a function that takes in a Binary Tree and returns the sum of its nodes' depths.

Each `BinaryTree` node has an integer `value`, a `left` child node, and a `right` child node. Children nodes can either be `BinaryTree` nodes themselves or `None` / `null`.

### Sample Input

```
tree =      1
           /  \
          2    3
         /  \  /  \
        4    5 6    7
       /  \
      8    9
```

### Sample Output

```
16
// The depth of the node with value 2 is 1.
// The depth of the node with value 3 is 1.
// The depth of the node with value 4 is 2.
// The depth of the node with value 5 is 2.
// Etc..
// Summing all of these depths yields 16.
```

# Hints

## Hint 1



As obvious as it may seem, to solve this question, you'll have to figure out how to compute the depth of any given node; once you know how to do that, you can compute all of the depths and add them up to obtain the desired output.

## Hint 2



To compute the depth of a given node, you need information about its position in the tree. Can you pass this information down from the node's parent?

## Hint 3



The depth of any node in the tree is equal to the depth of its parent node plus 1. By starting at the root node whose depth is 0, you can pass down to every node in the tree its respective depth, and you can implement the algorithm that does this and that sums up all of the depths either recursively or iteratively.

## Optimal Space & Time Complexity



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