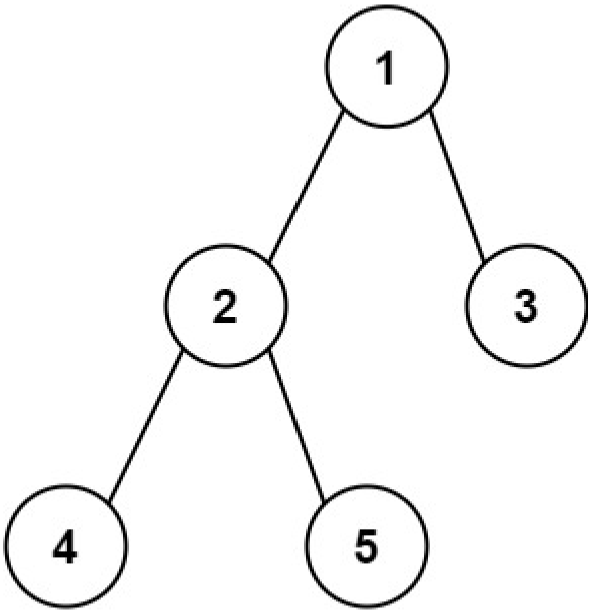
# 543. Diameter of Binary Tree

Given the root of a binary tree, return *the length of the **diameter** of the tree*.

The **diameter** of a binary tree is the **length** of the longest path between any two nodes in a tree. This path may or may not pass through the root.

The **length** of a path between two nodes is represented by the number of edges between them.

**Example 1:**



**Input:** root = [1,2,3,4,5]

**Output:** 3

**Explanation:** 3 is the length of the path [4,2,1,3] or [5,2,1,3].

**Example 2:**

**Input:** root = [1,2]

**Output:** 1

**Constraints:**

- The number of nodes in the tree is in the range [1, $10^4$].
- -100 <= Node.val <= 100

```
# Definition for a binary tree node.
# class TreeNode(object):
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution(object):
    def diameterOfBinaryTree(self, root):
        """
        :type root: TreeNode
        :rtype: int
        """
        diameter = [0]

        def helper(node):
            if not node:
                return 0;

            left = helper(node.left)
            right = helper(node.right)
            diameter[0] = max(diameter[0],left+right)

            return max(left,right)+1

        helper(root)
        return diameter[0]
```