



## 543. Diameter of Binary Tree

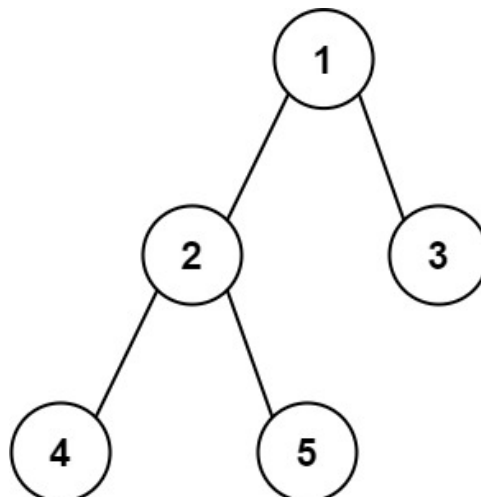
🕒 Created	@August 28, 2022 9:25 PM
⌵ Difficulty	Easy
≡ LC Url	<a href="https://leetcode.com/problems/diameter-of-binary-tree/">https://leetcode.com/problems/diameter-of-binary-tree/</a>
⌵ Importance	
⋮ Tag	DFS NEET Tree
≡ Video	<u>【HOT 100】9.二叉树的直径 Python3 递归也需要看清题意 - 二叉树的直径 - 力扣 (LeetCode)</u>

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, 104]`.
- `100 <= Node.val <= 100`

# Solution

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution:
    res = 0

    def diameterOfBinaryTree(self, root: Optional[TreeNode]) -> int:
        if not root:
            return 0

        self.dfs(root)

        return self.res

    def dfs(self, root):
        if not root:
            return 0
        left_depth = self.dfs(root.left)
        right_depth = self.dfs(root.right)
        # 将每个节点最大直径(左子树深度+右子树深度)当前最大值比较并取大者
        self.res = max(self.res, left_depth + right_depth)
        # 返回该节点为根的子树的深度
        return max(left_depth, right_depth) + 1
```

