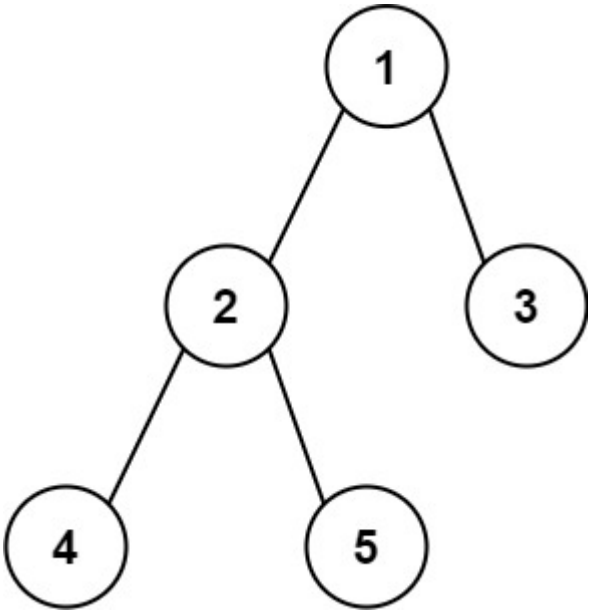
543. Diameter of Binary Tree(V.V.V.V.V Important)

> Very Important concept and question

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.




```
def diameterOfBinaryTree(self, root: TreeNode) -> int:
        if root.left is None and root.right is None:
            return 0
        ans = [0]
        self.helper(root,ans)
        return ans[0]



    def helper(self,root,res):
        if root is None:
            return -1
        ld = self.helper(root.left,res)
        rd = self.helper(root.right,res)
```

```
        res[0] = max(res[0], 2+ld + rd)
        return 1 + max(ld, rd)
```

```
class Solution:
    def diameterOfBinaryTree(self, root: Optional[TreeNode]) -> int:
        if root is None:
            return 0
        left = self.diameterOfBinaryTree(root.left)
        right = self.diameterOfBinaryTree(root.right)

        lH = self.height(root.left)
        rH = self.height(root.right)


        dia = lH+rH+2
        return max(max(left,right),dia)

    def height(self,root):
        return -1 if root is None else
max(self.height(root.left),self.height(root.right))+1
```