이진 트리에서 두 노드 간 가장 긴 경로의 길이를 출력하라.

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

```
# 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

1.dfs
class Solution:
    result = 0

    def diameterOfBinaryTree(self, root: TreeNode) -> int:
        def checkDepth(node, depth):
            if node == None:
                return depth - 1
            return max(checkDepth(node.left, depth + 1), checkDepth(node.right, depth + 1))

        def check(node):
            if node == None:
                return
            left = checkDepth(node.left, 1)
            right = checkDepth(node.right, 1)
            if self.result < left + right:
                self.result = left + right
            check(node.left)
            check(node.right)

        check(root)
        return self.result

class Solution:
    longest = 0
    def diameterOfBinaryTree(self, root: TreeNode) -> int:
        def dfs(node):
            if not node:
                return -1
            left = dfs(node.left)
            right = dfs(node.right)
            self.longest = max(self.longest, left + right + 2)
            return max(left, right) + 1
        dfs(root)
        return self.longest
```