동일한 값을 지닌 가장 긴 경로를 찾아라.

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

```
# 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 longestUnivaluePath(self, root: TreeNode) -> int:
        def dfs(node, val, count):
            if node == None or node.val != val:
                return count - 1
            return max(dfs(node.left, val, count + 1), dfs(node.right, val, count + 1))

        def findUniPath(node):
            if node == None:
                return
            left = dfs(node.left, node.val, 1)
            right = dfs(node.right, node.val, 1)
            if self.result < left + right:
                self.result = left + right
            findUniPath(node.left)
            findUniPath(node.right)

        findUniPath(root)
        return self.result


class Solution:
    result = 0

    def longestUnivaluePath(self, root: TreeNode) -> int:
        def dfs(node):
            if node == None:
                return 0
            left = dfs(node.left)
            right = dfs(node.right)
            if node.left and node.left.val == node.val:
                left += 1
            else:
                left = 0
            if node.right and node.right.val == node.val:
```

```
            right += 1
        else:
            right = 0
        self.result = max(self.result, left + right)
        return max(left, right)

    dfs(root)
    return self.result
```