1372. Longest ZigZag Path in a Binary Tree

You are given the `root` of a binary tree.

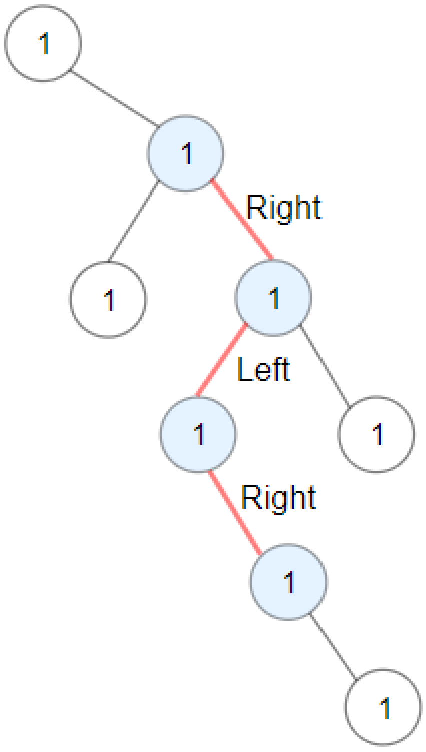
A ZigZag path for a binary tree is defined as follow:

- Choose **any** node in the binary tree and a direction (right or left).
- If the current direction is right, move to the right child of the current node; otherwise, move to the left child.
- Change the direction from right to left or from left to right.
- Repeat the second and third steps until you can't move in the tree.

Zigzag length is defined as the number of nodes visited - 1. (A single node has a length of 0).

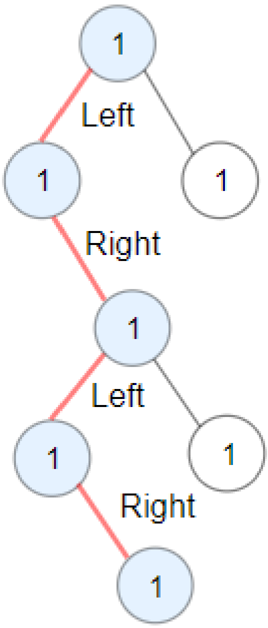
Return *the longest* ***ZigZag*** *path contained in that tree*.

Example 1:



```
Input: root = [1,null,1,1,1,null,null,1,1,null,1,null,null,null,1,null,1]
Output: 3
Explanation: Longest ZigZag path in blue nodes (right -> left -> right).
```

Example 2:



```
Input: root = [1,1,1,null,1,null,null,1,1,null,1]
Output: 4
Explanation: Longest ZigZag path in blue nodes (left -> right -> left ->
right).
```

Example 3:

```
Input: root = [1]
Output: 0
```

Constraints:

- The number of nodes in the tree is in the range `[1, 5 * 10<sup>4</sup>]`.
- `1 <= Node.val <= 100`

```
class Status:
    def __init__(self,a,b,c):
        self.forwardSlope = a
        self.backwardSlope = b
        self.maxLen = c
class Solution:
    def longestZigZag(self, root: Optional[TreeNode]) -> int:
        if root is None:
            return 0
        ans = self.longestZigZagHelper(root)
        return ans.maxLen



    def longestZigZagHelper(self,root):
        if root is None:
            return Status(-1,-1,0)
        leftStatus = self.longestZigZagHelper(root.left)
        rightStatus = self.longestZigZagHelper(root.right)
        maxLength = max(max(leftStatus.maxLen,rightStatus.maxLen),max(leftStatus.backwardSlope,rightStatus.forwardSlope)+1)
        forward = leftStatus.backwardSlope+1
        backward = rightStatus.forwardSlope+1
        return Status(forward,backward,maxLength)
```