class Solution(object):

def longestUnivaluePath(self, root):

"""

:type root: TreeNode

:rtype: int

"""

# Time: O(n)

# Space: O(n)

longest = [0]

def traverse(node):

if not node:

return 0

left\_len, right\_len = traverse(node.left), traverse(node.right)

left = (left\_len + 1) if node.left and node.left.val == node.val else 0

right = (right\_len + 1) if node.right and node.right.val == node.val else 0

longest[0] = max(longest[0], left + right)

return max(left, right)

traverse(root)

return longest[0]