**Justin\_Tree\_Recursion\_0687.**  **Longest Univalue Path**

**Concept:**

只需要算左右子樹的相同 value 的長度

**Code:**

# 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

class Solution:

def longestUnivaluePath(self, root: TreeNode) -> int:

self.result = 0

def arrow\_length(node):

if not node:

return 0

left\_length = arrow\_length(node.left)

right\_length = arrow\_length(node.right)

left\_arrow = right\_arrow = 0

if node.left and node.left.val == node.val:

left\_arrow = left\_length + 1

if node.right and node.right.val == node.val:

right\_arrow = right\_length + 1

self.result = max(self.result, left\_arrow + right\_arrow)

return max(left\_arrow, right\_arrow)

arrow\_length(root)

return self.result