**Justin\_Tree\_0993. Cousins in Binary Tree**

**Concept:**

如果左右子樹剛好存在 x, y 則回傳 False

持續找 x, y，並存著x, y 的深度

最後看 x, y 是否深度一樣

**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 isCousins(self, root: TreeNode, x: int, y: int) -> bool:

self.level\_x\_list = []

self.level\_y\_list = []

self.boolean = []

#print(type(root))

def find(node, level\_x, level\_y):

level\_x += 1

level\_y += 1

if node.left and node.right:

if (node.left.val == x and node.right.val == y) or (node.left.val == y and node.right.val == x):

self.boolean.append(0)

if node.left:

if node.left.val == x:

self.level\_x\_list.append(level\_x)

elif node.left.val == y:

self.level\_y\_list.append(level\_y)

find(node.left, level\_x, level\_y)

if node.right:

if node.right.val == x:

self.level\_x\_list.append(level\_x)

elif node.right.val == y:

self.level\_y\_list.append(level\_y)

find(node.right, level\_x, level\_y)

find(root, 0 ,0)