**Justin\_Tree\_Depth-firstSearch\_0110.**  **Balanced Binary Tree**

**Concept:**

一支副程式，算出該點子樹的深度

一支副程式，回傳左右子樹的深度是否相差 < 2，另外左右子樹子是否都為平衡

**Code:**

# Definition for a binary tree node.

# class TreeNode:

# def \_\_init\_\_(self, x):

# self.val = x

# self.left = None

# self.right = None

class Solution:

def find\_length(self, node):

if not node:

return -1

return 1 + max(self.find\_length(node.left), self.find\_length(node.right))

def isBalanced(self, node2) -> bool:

if not node2:

return True

return abs(self.find\_length(node2.left) - self.find\_length(node2.right)) < 2 and self.isBalanced(node2.left) and self.isBalanced(node2.right)