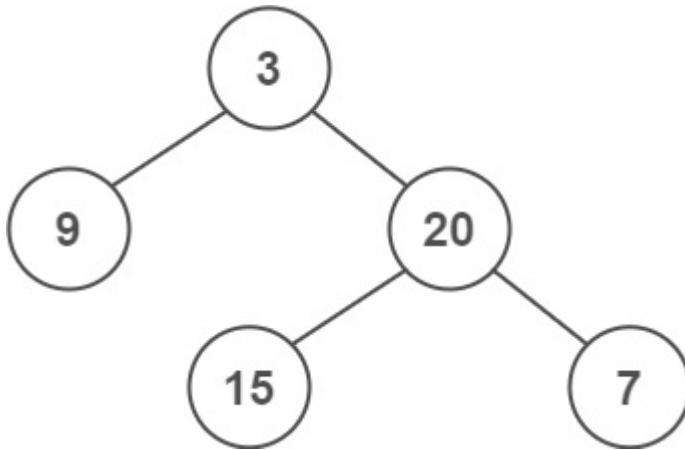


110. Balanced Binary Tree

Given a binary tree, determine if it is height-balanced.

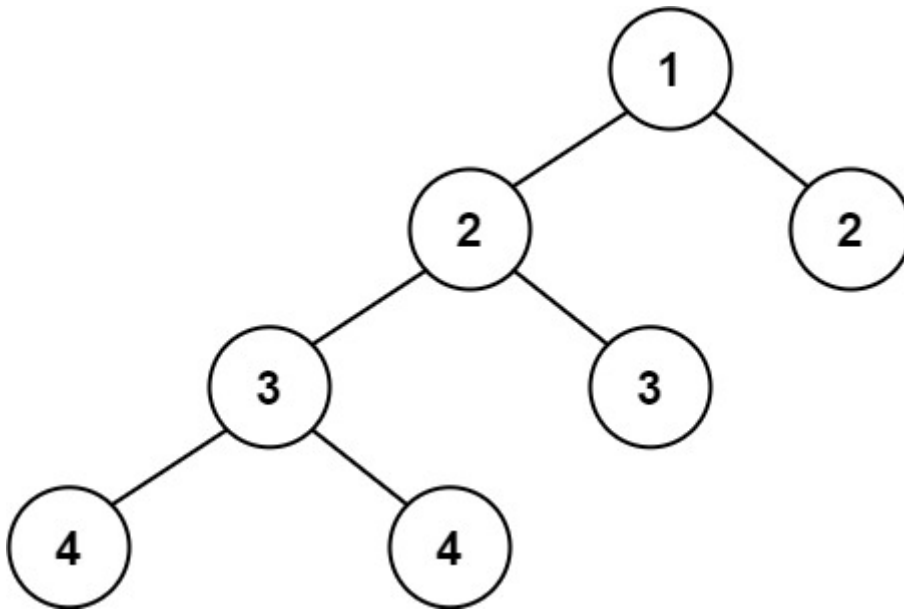
For this problem, a height-balanced binary tree is defined as:

a binary tree in which the left and right subtrees of every node differ in height by no more than 1.



Input: root = [3,9,20,null,null,15,7]

Output: true



Input: root = [1,2,2,3,3,null,null,4,4]

Output: false

Approach:1 ---> TC(n^2)

```
def isBalanced(self, root: TreeNode) -> bool:
    if root is None:
```

```

        return True

    lt = self.isBalanced(root.left)
    rt = self.isBalanced(root.right)
    return lt and rt and (abs(self.height(root.left) -
self.height(root.right)) <= 1)

def height(self, root):
    if root is None:
        return 0
    return 1 + max(self.height(root.left), self.height(root.right))

```

Approach:2 ---> Linear

```

def isBalanced(self, root: TreeNode) -> bool:
    if root is None:
        return True
    self.check = True
    self.helper(root)
    return self.check

def helper(self, root):
    if root is None:
        return 0
    lt = self.helper(root.left)
    rt = self.helper(root.right)
    if abs(lt - rt) > 1:
        self.check = False
    return 1 + max(lt, rt)

```