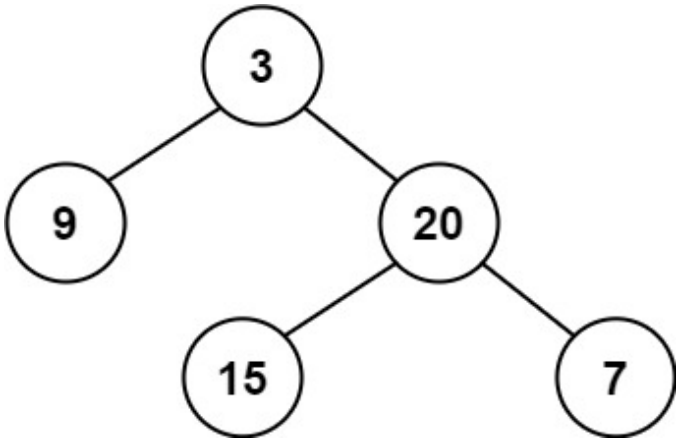
110. Balanced Binary Tree

Created	@November 26, 2022 10:52 AM
Difficulty	Easy
LC Url	https://leetcode.com/problems/balanced-binary-tree/
Importance	
Tag	NEET Tree
Video	

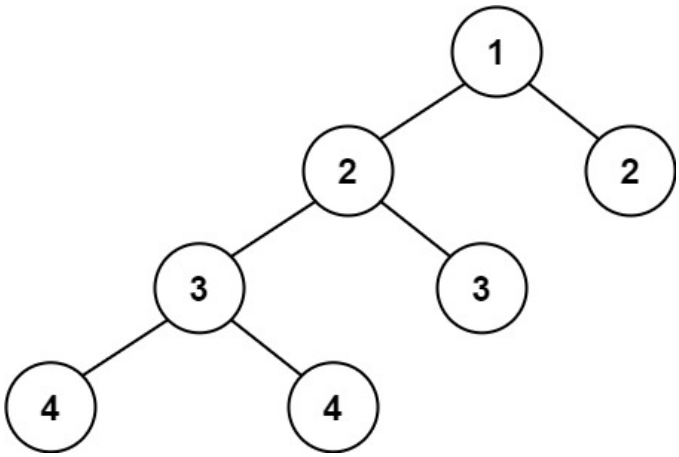
Given a binary tree, determine if it is **height-balanced**.

Example 1:




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

Example 2:




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

Example 3:

```
Input: root = []
Output: true
```

Constraints:

- The number of nodes in the tree is in the range `[0, 5000]`.
- `10 4 <= Node.val <= 10 4`

Solution

```
# 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:
    is_balanced = True

    def isBalanced(self, root: Optional[TreeNode]) -> bool:
        self.maxDepth(root)
        return self.is_balanced
```

```
    def maxDepth(self, root):
        if not root:
            return 0

        left_depth = self.maxDepth(root.left)
        right_depth = self.maxDepth(root.right)

        if abs(right_depth - left_depth) > 1:
            self.is_balanced = False

        return max(left_depth, right_depth) + 1
```