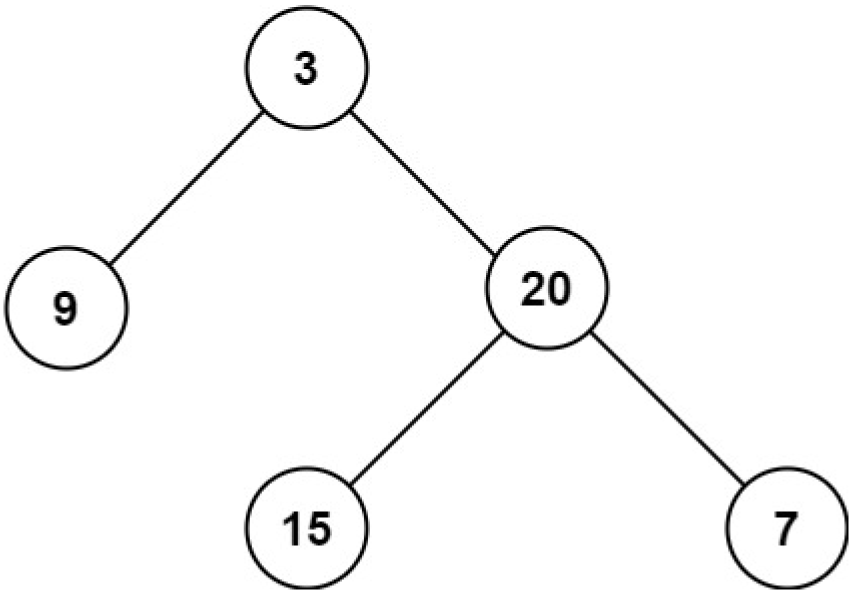
104. Maximum Depth of Binary Tree

Given the root of a binary tree, return *its maximum depth*.

A binary tree's **maximum depth** is the number of nodes along the longest path from the root node down to the farthest leaf node.

Example 1:



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

Output: 3

Example 2:

Input: root = [1,null,2]

Output: 2

Constraints:

- The number of nodes in the tree is in the range [0, 10^4].
- -100 <= Node.val <= 100

```
# Definition for a binary tree node.
# class TreeNode(object):
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution(object):
    def maxDepth(self, root):
        if not root:
            return 0
        return max(self.maxDepth(root.left),
self.maxDepth(root.right))+1
```

```
from collections import deque

class Solution:
    def maxDepth(self, root: TreeNode) -> int:
        if not root:
            return 0
        worklist = deque([root])
        num_node_level = 1
        levels = 0
        while worklist:
            node = worklist.popleft()
            if node.left:
                worklist.append(node.left)
            if node.right:
                worklist.append(node.right)
            num_node_level -= 1
            if num_node_level == 0:
                levels += 1
                num_node_level = len(worklist)

        return levels
```