

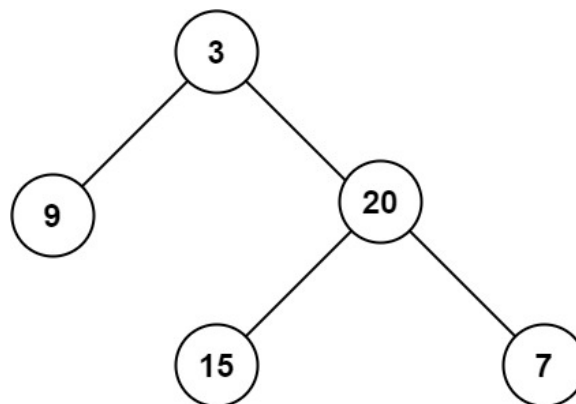
104. Maximum Depth of Binary Tree

🕒 Created	@September 10, 2022 7:52 PM
📌 Difficulty	Easy
📄 LC Url	https://leetcode.com/problems/maximum-depth-of-binary-tree/
📌 Importance	
🏷️ Tag	DFS NEET Recursion
📺 Video	https://maxming0.github.io/2020/12/01/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, 104]`.
- `-100 <= Node.val <= 100`

Solution

递归：深度为左右子树最大深度+1

```
class Solution:
    def maxDepth(self, root: TreeNode) -> int:
        return max(self.maxDepth(root.left), self.maxDepth(root.right)) + 1 if root else 0
```

BFS

非递归：从根开始bfs，每做一层，结果+1

```
# 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 maxDepth(self, root: Optional[TreeNode]) -> int:
        if not root:
            return 0

        queue = [root]
        depth = 0

        while queue:
            queue_temp = []
            for node in queue:
                if node.left:
                    queue_temp.append(node.left)
                if node.right:
                    queue_temp.append(node.right)
            queue = queue_temp
            depth += 1

        return depth
```

```
        queue = queue_temp
        depth += 1
    return depth
```