**Justin\_Tree\_Breadth-firstSearch\_0107. Binary Tree Level Order Traversal II**

**Concept:**

只要發現新的一層就加一個 []

找 node.left、node.right 並把 level + 1

最後的結果再反轉

**Code:**

# 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 levelOrderBottom(self, root: TreeNode) -> List[List[int]]:

result = []

if not root:

return result

def category(node, level):

if len(result) == level:

result.append([])

result[level].append(node.val)

if node.left:

category(node.left, level + 1)

if node.right:

category(node.right, level + 1)

category (root, 0)

return result[::-1]