# 题目

给定一棵二叉树的根节点 root ，请找出该二叉树中每一层的最大值。

示例1：

输入: root = [1,3,2,5,3,null,9]

输出: [1,3,9]

解释:

1

/ \

3 2

/ \ \

5 3 9

示例2：

输入: root = [1,2,3]

输出: [1,3]

解释:

1

/ \

2 3

示例3：

输入: root = [1]

输出: [1]

示例4：

输入: root = [1,null,2]

输出: [1,2]

解释:

  1

  \

  2

示例5：

输入: root = []

输出: []

提示：

二叉树的节点个数的范围是 [0,104]

-231 <= Node.val <= 231 - 1

注意：本题与主站 515 题相同：

https://leetcode-cn.com/problems/find-largest-value-in-each-tree-row/

# 分析

/\*\*

 \* Definition for a binary tree node.

 \* struct TreeNode {

 \*     int val;

 \*     TreeNode \*left;

 \*     TreeNode \*right;

 \*     TreeNode() : val(0), left(nullptr), right(nullptr) {}

 \*     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}

 \*     TreeNode(int x, TreeNode \*left, TreeNode \*right) : val(x), left(left), right(right) {}

 \* };

 \*/

class Solution {

public:

    vector<int> res;

    void dfs(TreeNode\* curNode, int level) {

        if(res.size() == level) res.push\_back(INT\_MIN);

        res[level] = max(res[level], curNode->val);

        if(curNode->left) dfs(curNode->left, level+1);

        if(curNode->right) dfs(curNode->right, level+1);

    }

    vector<int> largestValues(TreeNode\* root) {

        if(root == NULL) return res;

        dfs(root, 0);

        return res;

    }

};