LeetCode：1302. 层数最深叶子节点的和 题解：

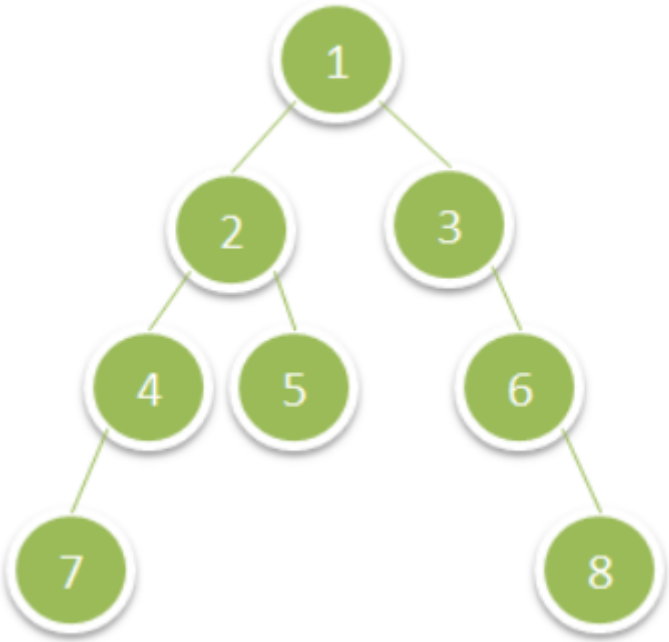
给你一棵二叉树，请你返回层数最深的叶子节点的和。

示例：
(详情请点击链接或者看下图)
输入：root = [1,2,3,4,5,null,6,7,null,null,null,null,8]
输出：15

提示：

树中节点数目在 1 到 10^4 之间。
每个节点的值在 1 到 100 之间。

来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/deepest-leaves-sum



思路简单介绍：

1. 暂时来说最优解是层序遍历。在此记录一下这种层序遍历的写法
2. 将第1层放入队列，开始循环，当队列为空时退出循环
3. 记录队列的长度len即该层的个数，ans初始化为0
4. 将队列中len个元素的左右孩子加入队列，并更新ans的大小，当len为0时退出循环
5. 当队列不为空，重复3，4
6. 队列为空，返回ans的值

show code:

```
/**
```

```
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    int deepestLeavesSum(TreeNode* root) {
        queue<TreeNode *> list;
        list.emplace(root);
        int len=0,ans=0;
        while(!list.empty()){
            len=list.size();
            ans=0;
            while(len--){
                TreeNode *temp=list.front();
                list.pop();
                if(temp->left!=NULL) list.emplace(temp->left);
                if(temp->right!=NULL) list.emplace(temp->right);
                ans+=temp->val;
            }
        }
        return ans;
    }
};
```