```
/*
给你一棵二叉树，请你返回层数最深的叶子节点的和。



示例：

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
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/
```

分析：

- BFS方法：使用层序遍历的方法, 记录下最后一层的累加和并返回即可。

方法一:C++_BFS

```
/**
 * 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)
        {
            int                 layer_sum = 0   ;
            queue<TreeNode *>   qtn             ;
            int                 num_1     = 0   ;
            int                 num_2     = 0   ;
            int                 i         = 0   ;
```

```
            TreeNode *          temp      = NULL;

            if(root)
            {
                qtn.push(root);
                num_2 = 1;
                num_1 = 0;

                while(num_2)
                {
                    num_1       = num_2 ;
                    layer_sum   = 0     ;
                    num_2       = 0     ;
                    for(i = 0 ; i < num_1 ;i++)
                    {
                        temp = qtn.front();
                        layer_sum += temp->val;
                        if(temp->left)
                        {
                            qtn.push(temp->left);
                            num_2++;
                        }

                        if(temp->right)
                        {
                            qtn.push(temp->right);
                            num_2++;
                        }
                        qtn.pop();
                    }
                }
            }
            return layer_sum;

        }
};


/*
执行结果:
通过
显示详情
执行用时 :52 ms, 在所有 C++ 提交中击败了72.83% 的用户
内存消耗 :31.9 MB, 在所有 C++ 提交中击败了100.00%的用户
*/
```

AlimyBreak
2020.01.08(2020第01周)