

199. Binary Tree Right Side View

题目描述: <https://leetcode.com/problems/binary-tree-right-side-view/>

给一二叉树，求其从右边看到节点。

例如:

```
      1             <---
     /  \
    2    3         <---
     \   \
     5    4         <---
```

You should return [1, 3, 4].

解题思路1:

按层遍历。只返回最右边的一列。

代码1:

```

/**
 * 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:
    vector<int> rightSideView(TreeNode* root) {
        if(root == NULL) return{};
        vector<vector<int> > res;
        queue<TreeNode*> q;
        q.push(root);
        while(!q.empty()) {
            int l = q.size();
            vector<int> t;
            while(l > 0) {
                l--;
                TreeNode* p = q.front();
                q.pop();
                t.push_back(p->val);
                if(p->left != NULL) {
                    q.push(p->left);
                }
                if(p->right != NULL) {
                    q.push(p->right);
                }
            }
            res.push_back(t);
        }
        vector<int> t;
        for(int i = 0; i < res.size(); i++) {
            t.push_back(res[i][res[i].size() - 1]);
        }
        return t;
    }
};

```

解题思路2:

从最右侧开始遍历，如果右边有节点则直接插入res，否则插入左边。

代码2:

```

/**
 * 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:
    void findRes(vector<int> &res, TreeNode* root, int level) {
        if(root == NULL) return;
        if(res.size() < level) res.push_back(root->val);
        findRes(res, root->right, level+1);
        findRes(res, root->left, level+1);
    }
    vector<int> rightSideView(TreeNode* root) {
        vector<int> res;
        findRes(res, root, 1);
        return res;
    }
};

```