#LeetCode 199

https://leetcode.com/problems/binary-tree-right-side-view/description/

Yifeng Zeng

#Description

199. Binary Tree Right Side View

#Idea Report

If I'm standing on the right side of a binary tree, I would see the right-most node of each level of the tree. So this becomes a problem to find the right-most node in each level of the tree. We can do a root-right-left preorder traversal and for each level we just record the first node that has been traversed. This can be achieved both using DFS or BFS.

Code

```
public class Solution {
    // DFS
    public List<Integer> rightSideView(TreeNode root) {
        List<Integer> res = new ArrayList<>();
        dfsHelper(res, root, 0);
        return res;
    }

    private void dfsHelper(List<Integer> res, TreeNode root, int level) {
        if (root == null) {
            return;
        }

        if (res.size() == level) {
            res.add(root.val);
        }

        dfsHelper(res, root.right, level + 1);
        dfsHelper(res, root.left, level + 1);
    }
}
```

Code

```
public class Solution {
    // BFS
    public List<Integer> rightSideView(TreeNode root) {
        List<Integer> res = new ArrayList<>();
        if (root == null) {
            return res;
        }

        Deque<TreeNode> q = new LinkedList<>();
        q.offer(root);
        int level = 0;

        while (!q.isEmpty()) {
            int size = q.size();
            for (int i = 0; i < size; i++) {
                TreeNode cur = q.poll();
                if (res.size() == level) {
                    res.add(cur.val);
                }
                if (cur.right != null) {
                    q.offer(cur.right);
                }
                if (cur.left != null) {
                    q.offer(cur.left);
                }
            }
            level++;
        }

        return res;
    }
}
```

#Summary

- Standard DFS/BFS operation in binary tree.