102. Binary Tree Level Order Traversal

link

利用queue来实现按层级遍历, 根据queue.size来确定每一层级应该有多少个元素, 因为每次add或者offer之前, size刚好等于本层元素个数.

```
    public List<List<Integer>> levelOrder(TreeNode root) {
        List<List<Integer>> res = new ArrayList();
        if(root == null) return res;
        Queue<TreeNode> queue = new LinkedList();
        queue.add(root);
        while(!queue.isEmpty()){
            int level = queue.size();
            List<Integer> list= new ArrayList();
            for(int i = 0; i < level; ++i){
                TreeNode node = queue.poll();
                if(node.left != null) queue.add(node.left);
                if(node.right != null) queue.add(node.right);
                list.add(node.val);
            }
            res.add(list);
        }
        return res;
    }
```

偷的preOrder DFS preOrder得复习一下.

```
public List<List<Integer>> levelOrder(TreeNode root) {
        List<List<Integer>> res = new ArrayList<List<Integer>>();
        levelHelper(res, root, 0);
        return res;
    }

    public void levelHelper(List<List<Integer>> res, TreeNode root, int height) {
        if (root == null) return;
        if (height >= res.size()) {
            res.add(new LinkedList<Integer>());
        }
        res.get(height).add(root.val);
        levelHelper(res, root.left, height+1);
        levelHelper(res, root.right, height+1);
    }
```