102. Binary Tree Level Order Traversal

Description | Hints | Submissions | Discuss | Solution

Pick One

Given a binary tree, return the *level order* traversal of its nodes' values. (ie, from left to right, level by level).

For example:
Given binary tree `[3,9,20,null,null,15,7]`,

```
    3
   / \
  9  20
    /  \
   15   7
```

return its level order traversal as:

```
[
  [3],
  [9,20],
  [15,7]
]
```

```java
/*
 * 树的层次遍历，用一个queue存储每层元素即可
 */
public class L102 {
    public class TreeNode{
        int val;
        TreeNode left;
        TreeNode right;
        public TreeNode(int x) {
            val = x;
        }
    }
    public List<List<Integer>> levelOrder(TreeNode root) {
        List<List<Integer>> list = new ArrayList<List<Integer>>();
        if(root == null)
            return list;
        Queue<TreeNode> queue = new LinkedList<>();
        queue.add(root);
        while(queue.size() != 0) {
            List<Integer> alist = new ArrayList<>();
            for(TreeNode child : queue)
                alist.add(child.val);
            list.add(alist);
            Queue<TreeNode> queue2 = queue;
            queue = new LinkedList<TreeNode>();
            for(TreeNode child : queue2) {
                if(child.left != null)
                    queue.add(child.left);
                if(child.right != null)
                    queue.add(child.right);
            }
        }
        return list;
    }
}
```