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

For example:

Given binary tree [3,9,20,null,null,15,7],

3 / \ 9 20 / \ 15 7

return its bottom-up level order traversal as:

[[15,7], [9,20], [3]] */

-
- 思想：
- (1) 利用 Queue 进行层次遍历
- (2) LinkedList具有addFirst方法，可以把元素直接插入到头部，这样就做到了逆序排布

```
public List<List<Integer>> levelOrderBottom(TreeNode root) {

    LinkedList<List<Integer>> result = new LinkedList<>();

    if (root == null) {
        return result;
    }

    Queue<TreeNode> queue = new LinkedList<>();

    queue.add(root);

    while (!queue.isEmpty()) {
        List<Integer> tempResult = new LinkedList<>();
        int size = queue.size();
        for (int i = 0; i < size; i++) {
            TreeNode currentNode = queue.poll();
            tempResult.add(currentNode.val);
            if (currentNode.left != null) {
                queue.add(currentNode.left);
            }
            if (currentNode.right != null) {
                queue.add(currentNode.right);
            }
        }
        result.addFirst(tempResult);
    }

    return result;
}
```