103. Binary Tree Zigzag Level Order Traversal

Description | Hints | Submissions | Discuss | Solution

Pick One

Given a binary tree, return the *zigzag level order* traversal of its nodes' values. (ie, from left to right, then right to left for the next level and alternate between).

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

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

return its zigzag level order traversal as:

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

```java
/*
 * 这道题目同样是BFS，用一个flag记录是否需要reverse，如果需要的话就把reverse的结果存储即可。
 */
public class L103 {
    public class TreeNode{
        int val;
        TreeNode left;
        TreeNode right;
        public TreeNode(int x) {
            val = x;
        }
    }
     public List<List<Integer>> zigzagLevelOrder(TreeNode root) {
         List<List<Integer>> res = new ArrayList<List<Integer>>();
         if(root == null)
             return res;
         LinkedList<TreeNode> queue = new LinkedList<TreeNode>();
         queue.add(root);
         int num = 0;
         boolean reverse = false;
         while (!queue.isEmpty()) {
            num = queue.size();
            ArrayList<Integer> levelres = new ArrayList<Integer>();
            for(int i = 0; i < num; i++) {
                TreeNode node = queue.poll();
                levelres.add(node.val);
                if(node.left != null)
                    queue.add(node.left);
                if(node.right != null)
                    queue.add(node.right);
            }
            if(reverse){
                Collections.reverse(levelres);//反转
                reverse = false;
            }else {
                reverse = true;
            }
            res.add(levelres);
        }
         return res;
     }
}
```