107. Binary Tree Level Order Traversal II

Description Hints Submissions Discuss Solution

Pick One

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).

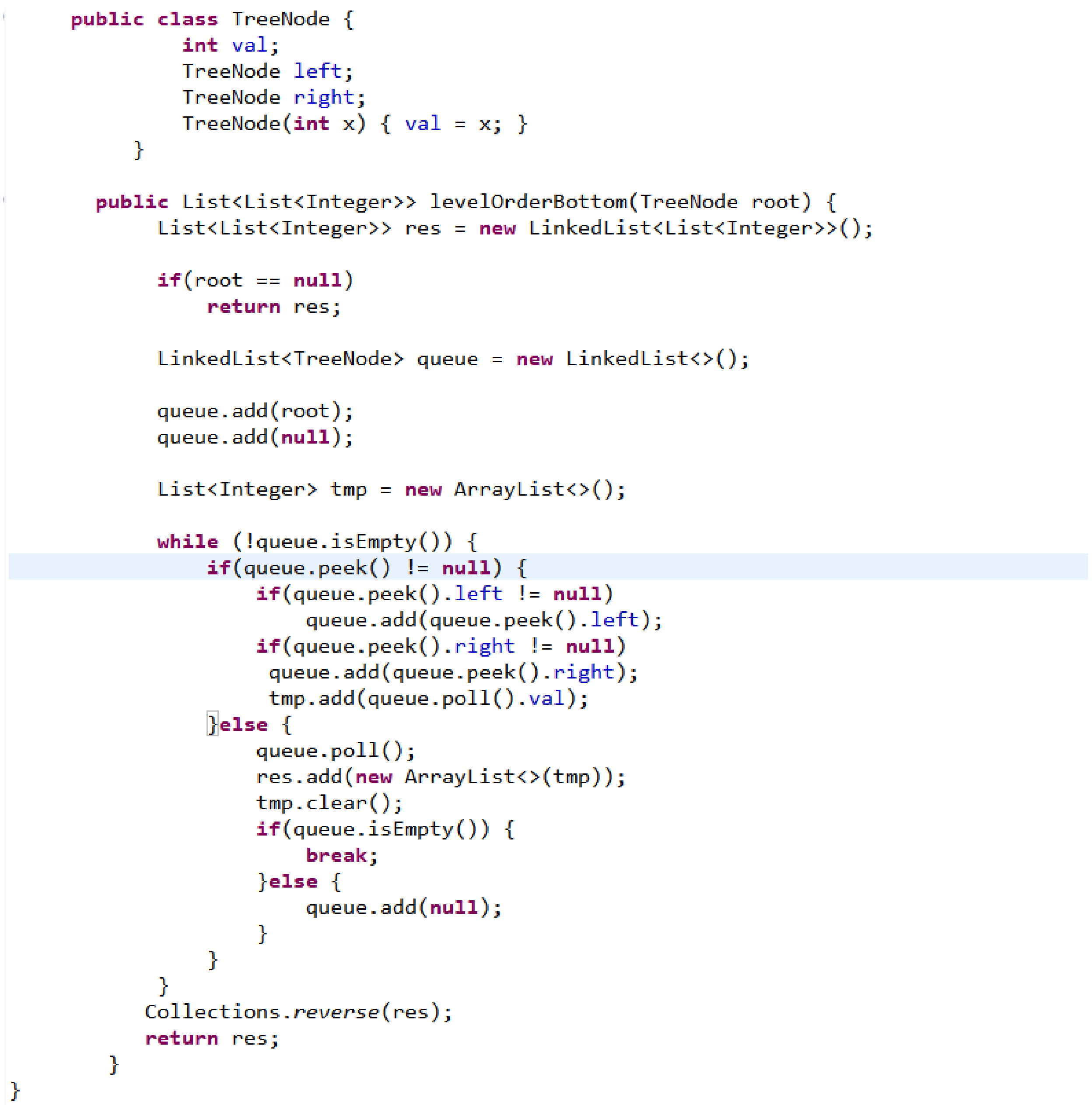
For example:
Given binary tree `[3,9,20,null,null,15,7]`,

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

return its bottom-up level order traversal as:

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

Seen this question in a real interview before? Yes No



```
     public class TreeNode {
              int val;
              TreeNode left;
              TreeNode right;
              TreeNode(int x) { val = x; }
          }

       public List<List<Integer>> levelOrderBottom(TreeNode root) {
            List<List<Integer>> res = new LinkedList<List<Integer>>();

            if(root == null)
                return res;

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

            queue.add(root);
            queue.add(null);

            List<Integer> tmp = new ArrayList<>();

            while (!queue.isEmpty()) {
                if(queue.peek() != null) {
                    if(queue.peek().left != null)
                        queue.add(queue.peek().left);
                    if(queue.peek().right != null)
                     queue.add(queue.peek().right);
                     tmp.add(queue.poll().val);
                }else {
                    queue.poll();
                    res.add(new ArrayList<>(tmp));
                    tmp.clear();
                    if(queue.isEmpty()) {
                        break;
                    }else {
                        queue.add(null);
                    }
                }
            }
           Collections.reverse(res);
           return res;
        }
}
```