Solution@22-08-23

515. Find Largest Value in Each Tree Row

```
package LeetCode;

import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import java.util.Queue;

public class LargetsValueInEachTreeRow {
   public List<Integer> largestValues(TreeNode root) {
      List<Integer> result = new ArrayList<>();
      if(root==null) return result;
      Queue<TreeNode> q = new LinkedList<>();
      q.offer(root);
      q.offer(null);
      int max = Integer.MIN_VALUE;
      while(!q.isEmpty()){
         TreeNode tmp = q.poll();
         if(tmp!=null){
            if(max<tmp.val) max = tmp.val;
            if(tmp.left!=null) q.offer(tmp.left);
            if(tmp.right!=null) q.offer(tmp.right);
         }else{
            result.add(max);
            if(!q.isEmpty()) q.offer(null);
            max=Integer.MIN_VALUE;
         }
      }
```

```
        return result;
    }
}
```

451. Sort Characters By Frequency

```
class Solution {
    public String frequencySort(String s) {
        Map<Character, Integer> map = new HashMap<>();
        for (char c: s.toCharArray()) {
            map.put(c, map.getOrDefault(c, 0) + 1);
        }
        List<Character>[] bucket = new List[s.length() + 1];
        for (Character key: map.keySet()) {
            int frequency = map.get(key);
            if (bucket[frequency] == null) {
                bucket[frequency] = new ArrayList<>();
            }
            bucket[frequency].add(key);
        }
        StringBuilder sb = new StringBuilder();
        for (int i = bucket.length - 1; i >= 0; i--) {
            if(bucket[i] != null) {
                for (char c: bucket[i]) {
                    for (int j = 0; j < map.get(c); j++) {
                        sb.append(c);
                    }
                }
            }
        }
        return sb.toString();
    }
```

```
}
```