406 根据身高重建队列

Label：贪心

假设有打乱顺序的一群人站成一个队列，数组 people 表示队列中一些人的属性（不一定按顺序）。每个 people[i] = [hi, ki] 表示第 i 个人的身高为 hi ，前面 正好 有 ki 个身高大于或等于 hi 的人。

请你重新构造并返回输入数组 people 所表示的队列。返回的队列应该格式化为数组 queue ，其中 queue[j] = [hj, kj] 是队列中第 j 个人的属性（queue[0] 是排在队列前面的人）。

输入：people = [[7,0],[4,4],[7,1],[5,0],[6,1],[5,2]]
输出：[[5,0],[7,0],[5,2],[6,1],[4,4],[7,1]]
解释：
编号为 0 的人身高为 5 ，没有身高更高或者相同的人排在他前面。
编号为 1 的人身高为 7 ，没有身高更高或者相同的人排在他前面。
编号为 2 的人身高为 5 ，有 2 个身高更高或者相同的人排在他前面，即编号为 0 和 1 的人。
编号为 3 的人身高为 6 ，有 1 个身高更高或者相同的人排在他前面，即编号为 1 的人。
编号为 4 的人身高为 4 ，有 4 个身高更高或者相同的人排在他前面，即编号为 0、1、2、3 的人。
编号为 5 的人身高为 7 ，有 1 个身高更高或者相同的人排在他前面，即编号为 1 的人。
因此 [[5,0],[7,0],[5,2],[6,1],[4,4],[7,1]] 是重新构造后的队列。

- 贪心（先排序，再插队）

```
class Solution {
    public int[][] reconstructQueue(int[][] people) {
        if (0 == people.length || 0 == people[0].length)
            return new int[0][0];
         //按照身高降序 K升序排序
        Arrays.sort(people, (array1, array2) -> {return array1[0] == array2[0] ?
array1[1] - array2[1] : array2[0] - array1[0];});

        List<int[]> list = new ArrayList<>();

        //K值定义为 排在h前面且身高大于或等于h的人数
        //因为从身高降序开始插入，此时所有人身高都大于等于h
        //因此K值即为需要插入的位置
        for (int[] i : people) {    // 当前这个人就是剩下为安排的人种最矮的，所以把他按照
k插入进去之后，不会影响之前已经插入人的位置
            list.add(i[1], i);// index, element
        }
        return list.toArray(new int[list.size()][]);

    }
}
```