1. 题目
2. 题干

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）。

1. 示例

示例 1：

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。

示例 2：

输入：people = [[6,0],[5,0],[4,0],[3,2],[2,2],[1,4]]

输出：[[4,0],[5,0],[2,2],[3,2],[1,4],[6,0]]

提示：

1 <= people.length <= 2000

0 <= hi <= 106

0 <= ki < people.length

题目数据确保队列可以被重建

1. 题解
2. 思路

题目很长，也不太好读懂，大概的意思是将原有的队列根据数组people中的规则重新排列，可以看到数组people中的每个人有两个属性，一个是身高，另一个是前面有几个人身高大于本人身高。也就是说，我们在重新构造队列时，要判断前面有几个人的身高大于本人。

像这种问题，通常首先要排序来简化题目的难度，如何排序呢？如果按照身高降序排列的话，那么之前插入数列中的人的身高一定都是比当前高的，那么当前插入的人就不会对之前的人的排序造成影响。当某个人要插入时，当前数列中的身高都是比他高的，我们直接判断当前数列的长度和原有的people数组中的第二个参数（有多少比他高的人排在他的前面）即可知道当前的人应该排在什么位置了。

1. 代码实现

Java；

class Solution {

    public int[][] reconstructQueue(int[][] people) {

        Arrays.sort(people, new Comparator<int[]>(){

            public int compare(int[] person1, int[] person2){

                if(person1[0] != person2[0]){

                    return person2[0] - person1[0];

                }else{

                    return person1[1] - person2[1];

                }

            }

        });

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

        for(int[] person : people){

            if(person[1] >= list.size()){

                list.add(person);

            }else{

                list.add(person[1], person);

            }

        }

        return list.toArray(new int[list.size()][]);

    }

}