

## 406. Queue Reconstruction by Height

---

题目描述: <https://leetcode.com/problems/queue-reconstruction-by-height/>

排队问题，每个人对应一个pair，全部人数给定一个pair数组，其中pair中第一个值代表此人的身高，第二个值代表在他前面有几个身高高于或者等于他的同学人数，要求按照这个要求排队，返回最终排队结果。

例如：

```
Input: [[7,0], [4,4], [7,1], [5,0], [6,1], [5,2]]
Output: [[5,0], [7,0], [5,2], [6,1], [4,4], [7,1]]
```

解题思路：

按照个头和前面人数排序成 [7,0],[7,1],[6,1],[5,0],[5,2],[4,4]

然后依次选择这样新插入的人，他的下标就等于比他高的人的人数。

1. 第一次插入[7,0], index = 0 => [7,0]
2. 第二次插入[7,1], index = 1 => [7,0],[7,1]
3. 第三次插入[6,1], index = 1 => [7,0],[6,1],[7,1]
4. 第四次插入[5,0], index = 0 => [5,0],[7,0],[6,1],[7,1]
5. 第五次插入[5,2], index = 2 => [5,0],[5,2],[7,0],[6,1],[7,1]
6. 第六次插入[4,4], index = 4 => [5,0],[5,2],[7,0],[6,1],[4,4],[7,1]

代码：

```
class Solution {
public:
    vector<pair<int, int>> reconstructQueue(vector<pair<int, int>>& people) {
        sort(people.begin(), people.end(), [](pair<int, int> p1, pair<int, int> p2)
        {
            if(p1.first > p2.first || (p1.first == p2.first && p1.second < p2.seco
nd)) {
                return true;
            }
            return false;
        });
        vector<pair<int, int> > res;
        for(int i = 0; i < people.size(); i++) {
            res.insert(res.begin()+people[i].second, people[i]);
        }
        return res;
    }
};
```