区间交集

题目描述：

给定一组闭区间，其中部分区间存在交集。任意两个给定区间的交集，称为公共区间（如：[1,2],[2,3]的公共区间为[2,2]，[3,5],[3,6]的公共区间为[3,5]）。公共区间之间若存在交集，则需要合并（如：[1,3],[3,5]区间存在交集[3,3]，须合并为[1,5]）。按升序排列输出合并后的区间列表。

输入描述：

一组区间列表，

区间数为 N：

0 <= N <= 1000；

区间元素为 X：

-10000 <= X <= 10000。

输出描述：

升序排列的合并后区间列表

补充说明：

1、区间元素均为数字，不考虑字母、符号等异常输入。

2、单个区间认定为无公共区间。

示例 1

输入：

```
0 3
1 3
3 5
3 6
```

输出：

```
1 5
```

说明：

[0,3]和[1,3]的公共区间为[1,3]，[0,3]和[3,5]的公共区间为[3,3]，[0,3]和[3,6]的公共区间为[3,3]，[1,3]和[3,5]的公共区间为[3,3]，[1,3]和[3,6]的公共区间为[3,3]，[3,5]和[3,6]的公共区间为[3,5]，公共区间列表为[[1,3],[3,3],[3,5]]；

[1,3],[3,3],[3,5]存在交集，须合并为[1,5]。

示例 2

输入：

```
0 3

1 4

4 7

5 8
```

输出：

```
1 3

4 4

5 7
```

说明：

示例 3

输入：

```
1 2
3 4
```

输出：

```
None
```

说明：

[1,2]和[3,4]无交集

```
#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;


int main() {

    int a, b;

    vector<pair<int, int>> ori;

    while (cin >> a >> b) {

        ori.emplace_back(a, b);

    }

    sort(ori.begin(), ori.end(),

        [](const auto& a, const auto& b) {return a.first < b.first;});

    vector<pair<int, int>> mid;

    for (int i = 0; i < ori.size(); i++) {

        for (int j = i + 1; j < ori.size(); j++) {

            if (ori[i].second < ori[j].first) break;

            mid.emplace_back(ori[j].first, min(ori[i].second, ori[j].second));

        }
```

```
    }
    if (mid.empty()) cout << "None" << endl;
    else {
        sort(mid.begin(), mid.end(),
            [](const auto& a, const auto& b) {return a.first < b.first;});
        vector<pair<int, int>> ans;
        int left = mid[0].first, right = mid[0].second;
        for (int i = 1; i < mid.size(); i++) {
            if (mid[i].first <= right) {
                right = max(right, mid[i].second);
            } else {
                ans.emplace_back(left, right);
                left = mid[i].first;
                right = mid[i].second;
            }
        }
        ans.emplace_back(left, right);
        for (const auto& p : ans) {
            cout << p.first << " " << p.second << endl;
        }
    }
}
```

```
// 64 位输出请用 printf("%lld")
```