1. Merge Intervals

Given a collection of intervals, merge all overlapping intervals.

**Example 1:**

Input: [[1,3],[2,6],[8,10],[15,18]]  
Output: [[1,6],[8,10],[15,18]]  
Explanation: Since intervals [1,3] and [2,6] overlaps, merge them into [1,6].

**Example 2:**

Input: [[1,4],[4,5]]  
Output: [[1,5]]  
Explanation: Intervals [1,4] and [4,5] are considered overlapping.

**NOTE:** input types have been changed on April 15, 2019. Please reset to default code definition to get new method signature.

**解**

解法1 先按照左端点排序，然后找右端点。注意cmp函数参数传引用，写在class里面时要用static函数

bool cmp(vector<int>&v1, vector<int>&v2){  
 return v1[0] < v2[0];  
}  
class Solution {  
public:  
 vector<vector<int>> merge(vector<vector<int>>& intervals) {  
 sort(intervals.begin(), intervals.end(), cmp);  
 vector<vector<int>>ans;  
 int i = 0;  
 int len = intervals.size();  
 while(i < len){  
 int j = i;  
 int max\_right = intervals[i][1];  
 while(j < len && max\_right >= intervals[j][0]){  
 max\_right = max(max\_right, intervals[j][1]);  
 j++;  
 }  
 ans.push\_back({intervals[i][0], max\_right});  
 i = j;  
 }  
 return ans;  
 }  
};

解法2 暴力。先构建无向图，然后按照图遍历算法求出覆盖