Meeting Rooms

Problem Description:
Given an array of meeting time intervals consisting of start and end times [[s1,e1],[s2,e2],...] (si < ei), determine if a person could attend all meetings.

For example,
Given [[0, 30],[5, 10],[15, 20]],
return false.
solve:
The idea is pretty simple: first we sort the intervals in the ascending order of start; then we check for the overlapping of each pair of neighboring intervals. If they do, then return false; after we finish all the checks and have not returned false, just return true.
Sorting takes O(nlogn) time and the overlapping checks take O(n) time, so this idea isO(nlogn) time in total.

```
class Solution {
public:
    bool canAttendMeetings(vector<Interval>& intervals) {
        sort(intervals.begin(), intervals.end(), compare);
        int n = intervals.size();
        for (int i = 0; i < n - 1; i++)
            if (overlap(intervals[i], intervals[i + 1]))
                return false;
        return true;
    }
private:
    static bool compare(Interval& interval1, Interval& interval2) {
        return interval1.start < interval2.start;
    }
    bool overlap(Interval& interval1, Interval& interval2) {
        return interval1.end > interval2.start;
    }
};
```