Meeting Rooms II

Given an array of meeting time intervals consisting of start and end times [[s1,e1],[s2,e2],...] (si < ei), find the minimum number of conference rooms required.

Example 1:

```
Input: [[0, 30],[5, 10],[15, 20]]
Output: 2
```

```
class Solution {
    public int minMeetingRooms(int[][] intervals) {

        if (intervals == null || intervals.length == 0)
            return 0;

        // sort the intervals by start time
        Arrays.sort(intervals, new Comparator<int[]>() {
            public int compare(int[] a, int[] b) {
                return a[0] - b[0];
            }
        });

        // use a min heap to track the minimum end time
        PriorityQueue<int[]> heap =
            new PriorityQueue<int[]>(intervals.length, new Comparator<int[]>() {
            public int compare(int[] a, int[] b) {
                return a[1] - b[1];
            }
        });

        // start with the first meeting, put it to a meeting room
        heap.offer(intervals[0]);

        for (int i = 1; i < intervals.length; i++) {

            // get the meeting room that finishes earliest
            int[] interval = heap.poll();

            if (intervals[i][0] >= interval[1]) {
                interval[1] = intervals[i][1]; // merge the interval
            } else {
                heap.offer(intervals[i]);
            }

            // don't forget to put the meeting room back
            heap.offer(interval);
        }

        return heap.size();
    }
}
```