435. Non-overlapping Intervals

QuestionEditorial Solution
My Submissions

- Total Accepted: **3107**
- Total Submissions: **7857**
- Difficulty: **Medium**
- Contributors: **love_FDU_llp**

Given a collection of intervals, find the minimum number of intervals you need to remove to make the rest of the intervals non-overlapping.

Note:

1. You may assume the interval's end point is always bigger than its start point.
2. Intervals like [1,2] and [2,3] have borders "touching" but they don't overlap each other.

Example 1:

```
Input: [ [1,2], [2,3], [3,4], [1,3] ]

Output: 1

Explanation: [1,3] can be removed and the rest of intervals are non-overlapping.
```

Example 2:

```
Input: [ [1,2], [1,2], [1,2] ]

Output: 2

Explanation: You need to remove two [1,2] to make the rest of intervals non-overlapping.
```

Example 3:

```
Input: [ [1,2], [2,3] ]

Output: 0

Explanation: You don't need to remove any of the intervals since they're already non-overlapping.
```

```java
import java.util.Arrays;
import java.util.Comparator;

public class OverlapIntervals {


public static class Interval {
     int start;
     int end;
     Interval() { start = 0; end = 0; }
     Interval(int s, int e) { start = s; end = e; }
}

public static int eraseOverlapIntervals(Interval[] intervals) {
     Arrays.sort(intervals,new C());
     int count=intervals.length;
     long lastindex = Long.MIN_VALUE;
     for(Interval i:intervals)
     {
          if(i.start>=lastindex)
          {
               lastindex = i.end;
               count--;
          }
     }
    return count;
}

static class C implements Comparator<Interval>
{
     @Override
     public int compare(Interval o1, Interval o2) {
     // TODO Auto-generated method stub
          return Integer.compare(o1.end,o2.end);
     }

}

public static void main(String[] args) {
// TODO Auto-generated method stub
//driver the problem
// [ [1,2], [2,3], [3,4], [1,3] ]
int arrays[][] = {{1,2},{2,3},{3,4},{1,3}};
Interval[] inter = new Interval[arrays.length];
for(int i=0;i<arrays.length;i++)
{
     inter[i] = new Interval(arrays[i][0],arrays[i][1]);
}
```

```
int ans = eraseOverlapIntervals(inter);
System.out.println(ans);
}

}
```