436. Find Right Interval

Add to List
QuestionEditorial Solution
My Submissions

- Total Accepted: **3576**
- Total Submissions: **8507**
- Difficulty: **Medium**
- Contributors: **love_FDU_llp**

Given a set of intervals, for each of the interval i, check if there exists an interval j whose start point is bigger than or equal to the end point of the interval i, which can be called that j is on the "right" of i.

For any interval i, you need to store the minimum interval j's index, which means that the interval j has the minimum start point to build the "right" relationship for interval i. If the interval j doesn't exist, store -1 for the interval i. Finally, you need output the stored value of each interval as an array.

Note:

1. You may assume the interval's end point is always bigger than its start point.
2. You may assume none of these intervals have the same start point.

Example 1:

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

Output: [-1]

Explanation: There is only one interval in the collection, so it outputs -1.
```

Example 2:

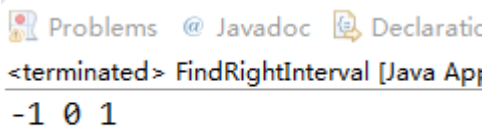
```
Input: [ [3,4], [2,3], [1,2] ]

Output: [-1, 0, 1]

Explanation: There is no satisfied "right" interval for [3,4].
For [2,3], the interval [3,4] has minimum-"right" start point;
For [1,2], the interval [2,3] has minimum-"right" start point.
```

Example 3:

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

Output: [-1, 2, -1]

Explanation: There is no satisfied "right" interval for [1,4] and [3,4].
For [2,3], the interval [3,4] has minimum-"right" start point.
```

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

public class FindRightInterval {

/**
 * Definition for an interval.
 * public class Interval {
 *     int start;
 *     int end;
 *     Interval() { start = 0; end = 0; }
 *     Interval(int s, int e) { start = s; end = e; }
 * }
 */
private static class Interval {
     int start;
```

```
    int end;
    Interval() { start = 0; end = 0; }
    Interval(int s, int e) { start = s; end = e; }
}

public static int[] findRightInterval(Interval[] intervals) {
    if(intervals.length==0) return new int[0];
    int n = intervals.length;
    Map<Interval, Integer> map = new HashMap<Interval,Integer>();
    for(int i=0;i<n;i++) map.put(intervals[i], i);
    Arrays.sort(intervals,new Comparator<Interval>() {

        @Override
        public int compare(Interval o1, Interval o2) {
        // TODO Auto-generated method stub
        return o1.start - o2.start;
        }
    });
    int[] res = new int[n];
    for(int i=0;i<n;i++)
    {
            int j=i+1;
            while(j<n && intervals[j].start < intervals[i].end)
j++;
            int idx = map.get(intervals[i]);
            int idx2= j<n ? map.get(intervals[j]) : -1;
            res[idx] = idx2;
    }
       return res;
}
public static void main(String[] args) {
// TODO Auto-generated method stub
    Interval[] test = new Interval[3];
    int[][] Array1 = {{3,4},{2,3},{1,2}};
    for(int i=0;i<test.length;++i)
    {
        Interval T = new Interval();
        T.start = Array1[i][0];
        T.end = Array1[i][1];
        test[i]=T;
    }
    int[] res = findRightInterval(test);
    for(int i=0;i<res.length;i++) System.out.printf("%d ",
res[i]);
}

}
```


Problems
Javadoc
Declaratic
<terminated> FindRightInterval [Java Ap

```
-1 0 1
```