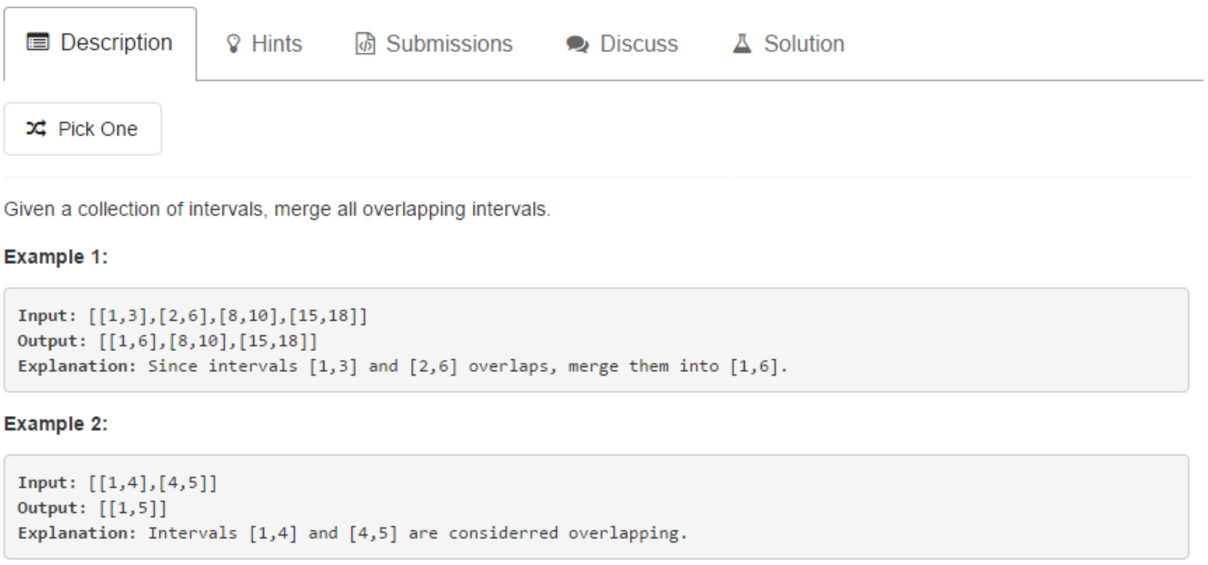
56. Merge Intervals



Description Hints Submissions Discuss Solution

Pick One

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 considerred overlapping.
```

```java
public class L56 {

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

    public List<Interval> merge(List<Interval> intervals) {
        List<Interval> result = new LinkedList<Interval>();

        if(intervals == null || intervals.size() < 1)
            return result;

        Collections.sort(intervals, new Comparator<Interval>() {
            public int compare(Interval o1, Interval o2) {
                return o1.start - o2.start;
            }
        });
      /*
       * 排序后，后一个元素（记为item）的start一定是不小于前一个(记为prev)start的，对于新添加的区间，如果item.start
       * 大于prev.end就说明这两个区间是分开的，要添加一个新区间，否则说明next.start在[prev.start,prev.end]内，
       * 则只要看next.end是否是大于prev.end，如果大于就要合并区间。
       */
        Interval prev = null;
        for (Interval items : intervals) {
            if(prev == null || prev.end < items.start) {
                result.add(items);
                prev = items;
            }else if (prev.end < items.end) {
                prev.end = items.end;
            }

        }
        return result;
    }
}
```