```
/**  In place
 * 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; }
 * }
 */
class Solution {
    public List<Interval> insert(List<Interval> ints, Interval ni) {
        //Collections.sort(ints, (a, b) -> a.start - b.start);

        int i = 0;
        while (i < ints.size() && ints.get(i).end < ni.start){
            i++;
        }

        int j = i;
        while (j < ints.size() && ints.get(j).start <= ni.end){
            ni = new Interval(Math.min(ints.get(j).start, ni.start), Math.max(ints.get(j).end, ni.end));
            j++;
        }

        for (int k = j-1; k >= i; k--) {
            ints.remove(k);
        }
        ints.add(i, ni);
        return ints;
    }
}


/**   Copy
 * 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; }
 * }
 */
public class Solution {
    public List<Interval> insert(List<Interval> ints, Interval ni) {
        List<Interval> list = new ArrayList<>();
        int i = 0;
        while (i < ints.size() && ints.get(i).end < ni.start){
            list.add(ints.get(i));
            i++;
        }

        while (i < ints.size() && ints.get(i).start <= ni.end){
            ni = new Interval(Math.min(ints.get(i).start, ni.start), Math.max(ints.get(i).end, ni.end));
            i++;
```

```
        }
        list.add(ni);

        while (i < ints.size()){
            list.add(ints.get(i++));
        }
        return list;
    }
}
```