A city's skyline is the outer contour of the silhouette formed by all the buildings in that city when viewed from a distance. Now suppose you are **given the locations and height of all the buildings** as shown on a cityscape photo (Figure A), write a program to **output the skyline** formed by these buildings collectively (Figure B).

[[](https://leetcode.com/static/images/problemset/skyline1.jpg)](https://leetcode.com/static/images/problemset/skyline1.jpg)[](https://leetcode.com/static/images/problemset/skyline2.jpg)

The geometric information of each building is represented by a triplet of integers [Li, Ri, Hi], where Li and Ri are the x coordinates of the left and right edge of the ith building, respectively, and Hi is its height. It is guaranteed that 0 ≤ Li, Ri ≤ INT\_MAX, 0 < Hi ≤ INT\_MAX, and Ri - Li > 0. You may assume all buildings are perfect rectangles grounded on an absolutely flat surface at height 0.

For instance, the dimensions of all buildings in Figure A are recorded as: [ [2 9 10], [3 7 15], [5 12 12], [15 20 10], [19 24 8] ] .

The output is a list of "**key points**" (red dots in Figure B) in the format of [ [x1,y1], [x2, y2], [x3, y3], ... ] that uniquely defines a skyline. **A key point is the left endpoint of a horizontal line segment**. Note that the last key point, where the rightmost building ends, is merely used to mark the termination of the skyline, and always has zero height. Also, the ground in between any two adjacent buildings should be considered part of the skyline contour.

For instance, the skyline in Figure B should be represented as:[ [2 10], [3 15], [7 12], [12 0], [15 10], [20 8], [24, 0] ].

**Notes:**

* The number of buildings in any input list is guaranteed to be in the range [0, 10000].
* The input list is already sorted in ascending order by the left x position Li.
* The output list must be sorted by the x position.
* There must be no consecutive horizontal lines of equal height in the output skyline. For instance, [...[2 3], [4 5], [7 5], [11 5], [12 7]...] is not acceptable; the three lines of height 5 should be merged into one in the final output as such: [...[2 3], [4 5], [12 7], ...]

问题分析：

利用divide and conquer

既然输入的数组已经按照left排序，那我们可以将数组一分为二（左右），分别计算得到左右矩形数组（得到的数组中的矩形也是按照left排序，并且互不交叉，构成天界线矩阵组），然后将两个数组融合。

所以算法的重点是如何将两个数组融合在一起。我们已经知道左右数组已经是互不交叉并且的已经排序的数组。所以我们每次只需要融合当前的左右数组的最左边的矩形：如果两个矩形互不交叉，那么我们可以直接将更左边的矩形加入返回数组；如果交叉，那么求出交叉得到的多个矩形加入返回数组。最后返回数组时，注意“去重”（就是首位相连的矩形如果高度一样就合并）。

最后我们根据得到的矩形数组构造天界线，实际上就是每个矩形的（left,height）。