Problem 218: The Skyline Problem

Problem Information

Difficulty: Hard
Acceptance Rate: 44.61%
Paid Only: No
Tags: Array, Divide and Conquer, Binary Indexed Tree, Segment Tree, Line Sweep, Sorting, Heap (Priority Queue), Ordered Set

Problem Description

A city's **skyline** is the outer contour of the silhouette formed by all the buildings in that city when viewed from a distance. Given the locations and heights of all the buildings, return _the**skyline** formed by these buildings collectively_.

The geometric information of each building is given in the array `buildings` where `buildings[i] = [lefti, righti, heighti]`:

* `lefti` is the x coordinate of the left edge of the `ith` building. * `righti` is the x coordinate of the right edge of the `ith` building. * `heighti` is the height of the `ith` building.

You may assume all buildings are perfect rectangles grounded on an absolutely flat surface at height `0`.

The **skyline** should be represented as a list of "key points" **sorted by their x-coordinate** in the form `[[x1,y1],[x2,y2],...]`. Each key point is the left endpoint of some horizontal segment in the skyline except the last point in the list, which always has a y-coordinate `0` and is used to mark the skyline's termination where the rightmost building ends. Any ground between the leftmost and rightmost buildings should be part of the skyline's contour.

Note: 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],...]`

Example 1:

![](https://assets.leetcode.com/uploads/2020/12/01/merged.jpg)

Input: buildings = [[2,9,10],[3,7,15],[5,12,12],[15,20,10],[19,24,8]] **Output:** [[2,10],[3,15],[7,12],[12,0],[15,10],[20,8],[24,0]] **Explanation:** Figure A shows the buildings of the input. Figure B shows the skyline formed by those buildings. The red points in figure B represent the key points in the output list.

Example 2:

Input: buildings = [[0,2,3],[2,5,3]] **Output:** [[0,3],[5,0]]

Constraints:

* `1 <= buildings.length <= 104` * `0 <= lefti < righti <= 231 - 1` * `1 <= heighti <= 231 - 1` * `buildings` is sorted by `lefti` in non-decreasing order.

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {

}
};
```

Java:

```
class Solution {
public List<List<Integer>> getSkyline(int[][] buildings) {

}
}
```

Python3:

```
class Solution:
def getSkyline(self, buildings: List[List[int]]) -> List[List[int]]:
```