6043. Count Number of Rectangles Containing Each Point

My Submissions (/contest/weekly-contest-290/problems/count-number-of-rectangles-containing-each-point/submissions/)

Back to Contest (/contest/weekly-contest-290/)

You are given a 2D integer array `rectangles` where `rectangles[i] = [l_i, h_i]` indicates that i^{th} rectangle has a length of l_i and a height of h_i. You are also given a 2D integer array `points` where `points[j] = [x_j, y_j]` is a point with coordinates (x_j, y_j).

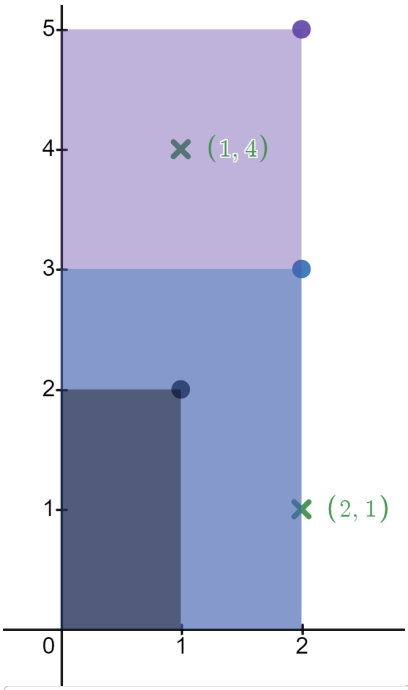
The i^{th} rectangle has its **bottom-left corner** point at the coordinates `(0, 0)` and its **top-right corner** point at (l_i, h_i).

Return *an integer array* `count` *of length* `points.length` *where* `count[j]` *is the number of rectangles that* ***contain*** *the* j^{th} *point.*

The i^{th} rectangle **contains** the j^{th} point if $0 <= x_j <= l_i$ and $0 <= y_j <= h_i$. Note that points that lie on the **edges** of a rectangle are also considered to be contained by that rectangle.

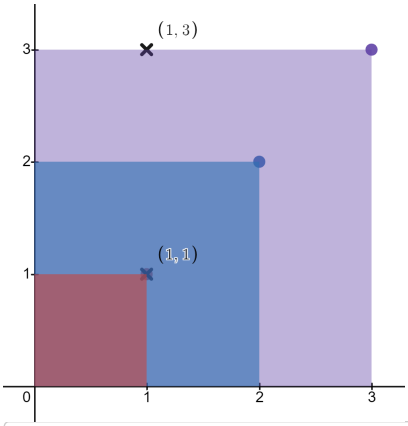
User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:




```
Input: rectangles = [[1,2],[2,3],[2,5]], points = [[2,1],[1,4]]
Output: [2,1]
Explanation:
The first rectangle contains no points.
The second rectangle contains only the point (2, 1).
The third rectangle contains the points (2, 1) and (1, 4).
The number of rectangles that contain the point (2, 1) is 2.
The number of rectangles that contain the point (1, 4) is 1.
Therefore, we return [2, 1].
```

Example 2:



Input: rectangles = [[1,1],[2,2],[3,3]], points = [[1,3],[1,1]]
Output: [1,3]
Explanation:
The first rectangle contains only the point (1, 1).
The second rectangle contains only the point (1, 1).
The third rectangle contains the points (1, 3) and (1, 1).
The number of rectangles that contain the point (1, 3) is 1.
The number of rectangles that contain the point (1, 1) is 3.
Therefore, we return [1, 3].

Constraints:

- 1 <= rectangles.length, points.length <= $5 * 10^4$
- rectangles[i].length == points[j].length == 2
- 1 <= l_i, x_j <= 10^9
- 1 <= h_i, y_j <= 100
- All the rectangles are **unique**.
- All the points are **unique**.

JavaScript

```
function Bisect() {
    return { insort_right, insort_left, bisect_left, bisect_right }
    function insort_right(a, x, lo = 0, hi = null) {
        lo = bisect_right(a, x, lo, hi);
        a.splice(lo, 0, x);
    }
    function bisect_right(a, x, lo = 0, hi = null) { // > upper_bound
        if (lo < 0) throw new Error('lo must be non-negative');
        if (hi == null) hi = a.length;
        while (lo < hi) {
            let mid = parseInt((lo + hi) / 2);
            a[mid] > x ? hi = mid : lo = mid + 1;
        }
        return lo;
    }
    function insort_left(a, x, lo = 0, hi = null) {
        lo = bisect_left(a, x, lo, hi);
        a.splice(lo, 0, x);
    }
    function bisect_left(a, x, lo = 0, hi = null) {
        if (lo < 0) throw new Error('lo must be non-negative');
        if (hi == null) hi = a.length;
        while (lo < hi) {
            let mid = parseInt((lo + hi) / 2);
            a[mid] < x ? lo = mid + 1 : hi = mid;
        }
        return lo;
    }
}

const initializeGraph = (n) => { let g = []; for (let i = 0; i < n; i++) { g.push([]); } return g; };

const countRectangles = (rectangles, points) => {
```

```
    let maxHeight = Math.max(...rectangles.map(e => e[1])), g = initializeGraph(maxHeight + 1), res = [], bi = new
Bisect();
    for (const [x, y] of rectangles) {
        g[y].push(x);
    }
    for (let i = 1; i <= maxHeight; i++) g[i].sort((x, y) => x - y);
    for (const [x, y] of points) {
        let cnt = 0;
        for (let i = y; i <= maxHeight; i++) {
            let idx = bi.bisect_left(g[i], x)
            cnt += g[i].length - idx;
        }
        res.push(cnt);
    }
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/686435758/) | More Details > (/submissions/detail/686435758/)

Share your acceptance!