5912. Most Beautiful Item for Each Query

My Submissions (/contest/biweekly-contest-65/problems/most-beautiful-item-for-each-query/submissions/)

Back to Contest (/contest/biweekly-contest-65/)

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

You are given a 2D integer array `items` where `items[i] = [`price_i`, `beauty_i`]` denotes the **price** and **beauty** of an item respectively.

You are also given a **0-indexed** integer array `queries`. For each `queries[j]`, you want to determine the **maximum beauty** of an item whose **price** is **less than or equal** to `queries[j]`. If no such item exists, then the answer to this query is `0`.

Return *an array* `answer` *of the same length as* `queries` *where* `answer[j]` *is the answer to the* j^{th} *query*.

Example 1:

```
Input: items = [[1,2],[3,2],[2,4],[5,6],[3,5]], queries = [1,2,3,4,5,6]
Output: [2,4,5,5,6,6]
Explanation:
- For queries[0]=1, [1,2] is the only item which has price <= 1. Hence, the answer for this query is 2.
- For queries[1]=2, the items which can be considered are [1,2] and [2,4].
  The maximum beauty among them is 4.
- For queries[2]=3 and queries[3]=4, the items which can be considered are [1,2], [3,2], [2,4], and [3,5].
  The maximum beauty among them is 5.
- For queries[4]=5 and queries[5]=6, all items can be considered.
  Hence, the answer for them is the maximum beauty of all items, i.e., 6.
```

Example 2:

```
Input: items = [[1,2],[1,2],[1,3],[1,4]], queries = [1]
Output: [4]
Explanation:
The price of every item is equal to 1, so we choose the item with the maximum beauty 4.
Note that multiple items can have the same price and/or beauty.
```

Example 3:

```
Input: items = [[10,1000]], queries = [5]
Output: [0]
Explanation:
No item has a price less than or equal to 5, so no item can be chosen.
Hence, the answer to the query is 0.
```

Constraints:

- `1 <= items.length, queries.length <=` 10^5
- `items[i].length == 2`
- `1 <=` price_i, beauty_i, `queries[j] <=` 10^9

JavaScript

```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);
            x < a[mid] ? 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) { // >= lower_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 ? lo = mid + 1 : hi = mid;
        }
        return lo;
    }
}

const maximumBeauty = (a, queries) => {
    let n = a.length;
    a.sort((x, y) => x[0] - y[0]);
    let ia = a.map(x => x[0]);
    let pre = preMax(a, n);
    // pr(a);
    // pr(ia);
    // pr(pre);
    let bi = new Bisect();
    let res = [];
    for (const q of queries) {
        let idx = bi.bisect_right(ia, q);
        idx--;
        // pr("q", q, "idx", idx, "last", a[idx], a.slice(0, idx + 1))
        res.push(pre[idx] || 0);
    }
    return res;
};

const preMax = (a, n) => {
    let pre = [], max = 0;
    for (let i = 0; i < n; i++) {
        if (a[i][1] > max) max = a[i][1];
        pre.push(max);
    }
    return pre;
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/586559887/) More Details (/submissions/detail/586559887/)

Share your acceptance!

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)