6096. Successful Pairs of Spells and Potions

My Submissions (/contest/biweekly-contest-80/problems/successful-pairs-of-spells-and-potions/submissions/)

Back to Contest (/contest/biweekly-contest-80/)

You are given two positive integer arrays `spells` and `potions`, of length `n` and `m` respectively, where `spells[i]` represents the strength of the ith spell and `potions[j]` represents the strength of the jth potion.

You are also given an integer `success`. A spell and potion pair is considered **successful** if the **product** of their strengths is **at least** `success`.

Return *an integer array* `pairs` *of length* `n` *where* `pairs[i]` *is the number of* **potions** *that will form a successful pair with the* i^{th} *spell.*

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

Example 1:

```
Input: spells = [5,1,3], potions = [1,2,3,4,5], success = 7
Output: [4,0,3]
Explanation:
- 0ᵗʰ spell: 5 * [1,2,3,4,5] = [5,10,15,20,25]. 4 pairs are successful.
- 1ˢᵗ spell: 1 * [1,2,3,4,5] = [1,2,3,4,5]. 0 pairs are successful.
- 2ⁿᵈ spell: 3 * [1,2,3,4,5] = [3,6,9,12,15]. 3 pairs are successful.
Thus, [4,0,3] is returned.
```

Example 2:

```
Input: spells = [3,1,2], potions = [8,5,8], success = 16
Output: [2,0,2]
Explanation:
- 0ᵗʰ spell: 3 * [8,5,8] = [24,15,24]. 2 pairs are successful.
- 1ˢᵗ spell: 1 * [8,5,8] = [8,5,8]. 0 pairs are successful.
- 2ⁿᵈ spell: 2 * [8,5,8] = [16,10,16]. 2 pairs are successful.
Thus, [2,0,2] is returned.
```

Constraints:

- `n == spells.length`
- `m == potions.length`
- $1 <= n, m <= 10^5$
- $1 <= spells[i], potions[i] <= 10^5$
- $1 <= success <= 10^{10}$

JavaScript ▾

```javascript
 1 ▾ function Bisect() {
 2       return { insert_right, insert_left, bisect_left, bisect_right }
 3 ▾     function insert_right(a, x, lo = 0, hi = null) {
 4           lo = bisect_right(a, x, lo, hi);
 5           a.splice(lo, 0, x);
 6       }
 7 ▾     function bisect_right(a, x, lo = 0, hi = null) { // > upper_bound
 8           if (lo < 0) throw new Error('lo must be non-negative');
 9           if (hi == null) hi = a.length;
10 ▾         while (lo < hi) {
11               let mid = parseInt((lo + hi) / 2);
12               a[mid] > x ? hi = mid : lo = mid + 1;
13           }
14           return lo;
15       }
16 ▾     function insert_left(a, x, lo = 0, hi = null) {
17           lo = bisect_left(a, x, lo, hi);
18           a.splice(lo, 0, x);
19       }
20 ▾     function bisect_left(a, x, lo = 0, hi = null) { // >= lower_bound
21           if (lo < 0) throw new Error('lo must be non-negative');
```

```
22          if (hi == null) hi = a.length;
23 ▾        while (lo < hi) {
24              let mid = parseInt((lo + hi) / 2);
25              a[mid] < x ? lo = mid + 1 : hi = mid;
26          }
27          return lo;
28      }
29  }
30
31 ▾ const successfulPairs = (a, b, p) => {
32      a = a.map((x, i) => [x, i]);
33      b.sort((x, y) => x - y);
34      let bi = new Bisect(), res = Array(a.length).fill(0);
35 ▾    for (const [x, i] of a) {
36          let min = Math.ceil(p / x), idx = bi.bisect_left(b, min), cnt = b.length - idx;
37          res[i] = cnt;
38      }
39      return res;
40  };
```

☐ **Custom Testcase**   (Use Example Testcases)

(▶ Run)   (☁ Submit)

Submission Result: Accepted (/submissions/detail/719660718/) ❓   (More Details ❯ (/submissions/detail/719660718/))

Share your acceptance!