

Problem 2300: Successful Pairs of Spells and Potions

Problem Information

Difficulty: **Medium**

Acceptance Rate: 49.26%

Paid Only: No

Tags: Array, Two Pointers, Binary Search, Sorting

Problem Description

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

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 <= 105` * `1 <= spells[i], potions[i] <= 105` * `1 <= success <= 1010`
```

Code Snippets

C++:

```
class Solution {
public:
    vector<int> successfulPairs(vector<int>& spells, vector<int>& potions, long
    long success) {

    }
};
```

Java:

```
class Solution {
    public int[] successfulPairs(int[] spells, int[] potions, long success) {

    }
}
```

Python3:

```
class Solution:
    def successfulPairs(self, spells: List[int], potions: List[int], success:
    int) -> List[int]:
```