Fruits Into Baskets III

You are given two arrays of integers, fruits and baskets, each of length n, where fruits[i] represents the **quantity** of the i^{th} type of fruit, and baskets[j] represents the **capacity** of the j^{th} basket.

From left to right, place the fruits according to these rules:

- Each fruit type must be placed in the **leftmost available basket** with a capacity **greater than or equal** to the quantity of that fruit type.
- Each basket can hold **only one** type of fruit.
- If a fruit type **cannot be placed** in any basket, it remains **unplaced**.

Return the number of fruit types that remain unplaced after all possible allocations are made.

Example 1:

Input: fruits = [4,2,5], baskets = [3,5,4]

Output: 1

Explanation:

- fruits[0] = 4 is placed in baskets[1] = 5.
- fruits[1] = 2 is placed in baskets[0] = 3.
- fruits[2] = 5 cannot be placed in baskets[2] = 4.

Since one fruit type remains unplaced, we return 1.

Example 2:

Input: fruits = [3,6,1], baskets = [6,4,7]

Output: 0

Explanation:

- fruits[0] = 3 is placed in baskets[0] = 6.
- fruits[1] = 6 cannot be placed in baskets[1] = 4 (insufficient capacity) but can be placed in the next available basket, baskets[2] = 7.
- fruits[2] = 1 is placed in baskets[1] = 4.

Since all fruits are successfully placed, we return 0.

Constraints:

- n == fruits.length == baskets.length
- 1 <= n <= 10^5
- 1 <= fruits[i], baskets[i] <= 10^9