You have n tasks and m workers. Each task has a strength requirement stored in a **0-indexed** integer array tasks, with the ith task requiring tasks[i] strength to complete. The strength of each worker is stored in a **0-indexed** integer array workers, with the jth worker having workers[j] strength. Each worker can only be assigned to a **single** task and must have a strength **greater than or equal** to the task's strength requirement (i.e., workers[j] >= tasks[i]).

Additionally, you have pills magical pills that will **increase a worker's strength** by strength. You can decide which workers receive the magical pills, however, you may only give each worker **at most one** magical pill.

Given the **0-indexed** integer arrays tasks and workers and the integers pills and strength, return *the* ***maximum*** *number of tasks that can be completed.*

**Example 1:**

Input: tasks = [3,2,1], workers = [0,3,3], pills = 1, strength = 1  
Output: 3  
Explanation:  
We can assign the magical pill and tasks as follows:  
- Give the magical pill to worker 0.  
- Assign worker 0 to task 2 (0 + 1 >= 1)  
- Assign worker 1 to task 1 (3 >= 2)  
- Assign worker 2 to task 0 (3 >= 3)

**Example 2:**

Input: tasks = [5,4], workers = [0,0,0], pills = 1, strength = 5  
Output: 1  
Explanation:  
We can assign the magical pill and tasks as follows:  
- Give the magical pill to worker 0.  
- Assign worker 0 to task 0 (0 + 5 >= 5)

**Example 3:**

Input: tasks = [10,15,30], workers = [0,10,10,10,10], pills = 3, strength = 10  
Output: 2  
Explanation:  
We can assign the magical pills and tasks as follows:  
- Give the magical pill to worker 0 and worker 1.  
- Assign worker 0 to task 0 (0 + 10 >= 10)  
- Assign worker 1 to task 1 (10 + 10 >= 15)  
The last pill is not given because it will not make any worker strong enough for the last task.

**Constraints:**

* n == tasks.length
* m == workers.length
* 1 <= n, m <= 5 \* 104
* 0 <= pills <= m
* 0 <= tasks[i], workers[j], strength <= 109