Problem 2071: Maximum Number of Tasks You Can Assign

Problem Information

Difficulty: Hard
Acceptance Rate: 50.27%
Paid Only: No
Tags: Array, Two Pointers, Binary Search, Greedy, Queue, Sorting, Monotonic Queue

Problem Description

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`

Code Snippets

C++:

```
class Solution {
public:
int maxTaskAssign(vector<int>& tasks, vector<int>& workers, int pills, int
strength) {

}
};
```

Java:

```
class Solution {
public int maxTaskAssign(int[] tasks, int[] workers, int pills, int strength)
{

}
}
```

Python3:

```
class Solution:
def maxTaskAssign(self, tasks: List[int], workers: List[int], pills: int,
strength: int) -> int:
```