

2323. Find Minimum Time to Finish All Jobs II

Description

You are given two **0-indexed** integer arrays `jobs` and `workers` of **equal** length, where `jobs[i]` is the amount of time needed to complete the i^{th} job, and `workers[j]` is the amount of time the j^{th} worker can work each day.

Each job should be assigned to **exactly** one worker, such that each worker completes **exactly** one job.

Return *the minimum number of days needed to complete all the jobs after assignment*.

Example 1:

Input: `jobs = [5,2,4]`, `workers = [1,7,5]`

Output: 2

Explanation:

- Assign the 2nd worker to the 0th job. It takes them 1 day to finish the job.
- Assign the 0th worker to the 1st job. It takes them 2 days to finish the job.
- Assign the 1st worker to the 2nd job. It takes them 1 day to finish the job.

It takes 2 days for all the jobs to be completed, so return 2.

It can be proven that 2 days is the minimum number of days needed.

Example 2:

Input: `jobs = [3,18,15,9]`, `workers = [6,5,1,3]`

Output: 3

Explanation:

- Assign the 2nd worker to the 0th job. It takes them 3 days to finish the job.
- Assign the 0th worker to the 1st job. It takes them 3 days to finish the job.
- Assign the 1st worker to the 2nd job. It takes them 3 days to finish the job.
- Assign the 3rd worker to the 3rd job. It takes them 3 days to finish the job.

It takes 3 days for all the jobs to be completed, so return 3.

It can be proven that 3 days is the minimum number of days needed.

Constraints:

- `n == jobs.length == workers.length`
- `1 <= n <= 105`
- `1 <= jobs[i], workers[i] <= 105`

