Problem 3476: Maximize Profit from Task Assignment

Problem Information

Difficulty: Medium
Acceptance Rate: 65.76%
Paid Only: Yes
Tags: Array, Greedy, Sorting, Heap (Priority Queue)

Problem Description

You are given an integer array `workers`, where `workers[i]` represents the skill level of the `ith` worker. You are also given a 2D integer array `tasks`, where:

* `tasks[i][0]` represents the skill requirement needed to complete the task. * `tasks[i][1]` represents the profit earned from completing the task.

Each worker can complete **at most** one task, and they can only take a task if their skill level is **equal** to the task's skill requirement. An **additional** worker joins today who can take up _any_ task, **regardless** of the skill requirement.

Return the **maximum** total profit that can be earned by optimally assigning the tasks to the workers.

Example 1:

Input: workers = [1,2,3,4,5], tasks = [[1,100],[2,400],[3,100],[3,400]]

Output: 1000

Explanation:

* Worker 0 completes task 0. * Worker 1 completes task 1. * Worker 2 completes task 3. * The additional worker completes task 2.

Example 2:

Input: workers = [10,10000,100000000], tasks = [[1,100]]

Output: 100

Explanation:

Since no worker matches the skill requirement, only the additional worker can complete task 0.

Example 3:

Input: workers = [7], tasks = [[3,3],[3,3]]

Output: 3

Explanation:

The additional worker completes task 1. Worker 0 cannot work since no task has a skill requirement of 7.

Constraints:

* `1 <= workers.length <= 105` * `1 <= workers[i] <= 109` * `1 <= tasks.length <= 105` * `tasks[i].length == 2` * `1 <= tasks[i][0], tasks[i][1] <= 109`

Code Snippets

C++:

```
class Solution {
public:
long long maxProfit(vector<int>& workers, vector<vector<int>>& tasks) {

}
};
```

Java:

```
class Solution {
public long maxProfit(int[] workers, int[][] tasks) {

}
}
```

Python3:

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