Problem 826: Most Profit Assigning Work

Problem Information

Difficulty: Medium
Acceptance Rate: 56.07%
Paid Only: No
Tags: Array, Two Pointers, Binary Search, Greedy, Sorting

Problem Description

You have `n` jobs and `m` workers. You are given three arrays: `difficulty`, `profit`, and `worker` where:

* `difficulty[i]` and `profit[i]` are the difficulty and the profit of the `ith` job, and * `worker[j]` is the ability of `jth` worker (i.e., the `jth` worker can only complete a job with difficulty at most `worker[j]`).

Every worker can be assigned **at most one job** , but one job can be **completed multiple times**.

* For example, if three workers attempt the same job that pays `$1`, then the total profit will be `$3`. If a worker cannot complete any job, their profit is `$0`.

Return the maximum profit we can achieve after assigning the workers to the jobs.

Example 1:

Input: difficulty = [2,4,6,8,10], profit = [10,20,30,40,50], worker = [4,5,6,7] **Output:** 100 **Explanation:** Workers are assigned jobs of difficulty [4,4,6,6] and they get a profit of [20,20,30,30] separately.

Example 2:

Input: difficulty = [85,47,57], profit = [24,66,99], worker = [40,25,25] **Output:** 0

Constraints:

* `n == difficulty.length` * `n == profit.length` * `m == worker.length` * `1 <= n, m <= 104` * `1 <= difficulty[i], profit[i], worker[i] <= 105`

Code Snippets

C++:

```
class Solution {
public:
int maxProfitAssignment(vector<int>& difficulty, vector<int>& profit,
vector<int>& worker) {

}
};
```

Java:

```
class Solution {
public int maxProfitAssignment(int[] difficulty, int[] profit, int[] worker)
{

}
}
```

Python3:

```
class Solution:
def maxProfitAssignment(self, difficulty: List[int], profit: List[int],
worker: List[int]) -> int:
```