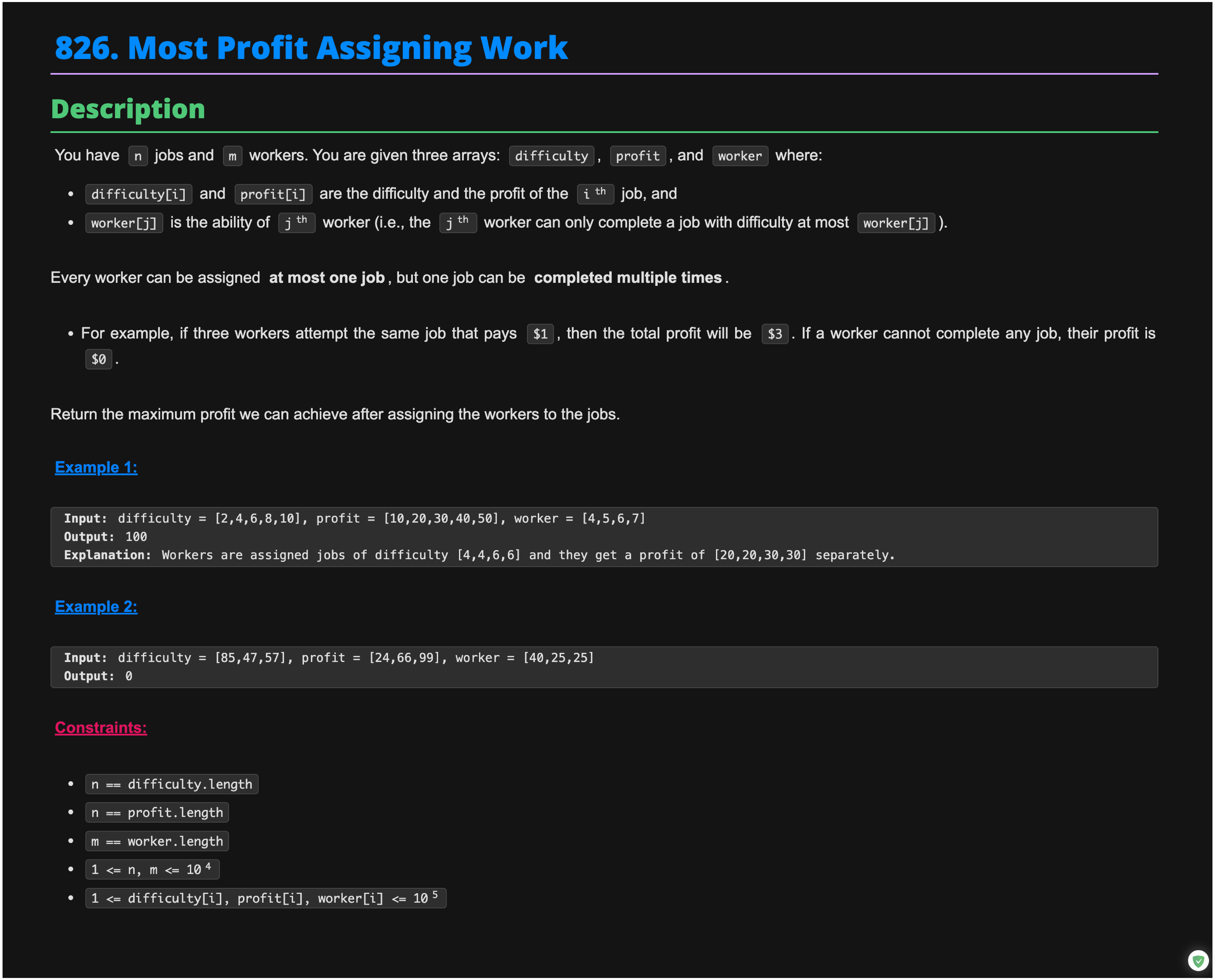

826. Most Profit Assigning Work

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 i^{th} job, and
- `worker[j]` is the ability of j^{th} worker (i.e., the j^{th} 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 <=` 10^4
- `1 <= difficulty[i], profit[i], worker[i] <=` 10^5