Task 2 - Data center

prepared for EGOI 2022 by Cheng Zhong

G is an internet company that runs many services and has many data centers worldwide. Each data center has some machines. Each service requires some machines to run on, and for security and redundancy reasons, it needs to be duplicated in multiple data centers.

While G plans to launch a new service that requires M machines and D duplicates, it sorts the data centers by their current free machines in descending order, and reserves M machines in each of the top D data centers.

Please calculate, after launching some services, the remaining free machines in the data centers.

Input

The first line contains N and S, the number of data centers and the number of new services to be launched.

The next line contains N numbers, represent the numbers of free machines in the N data centers at the beginning.

In the following S lines, each line corresponds to a new service: it contains two numbers M and D, indicate the number of machines and the number of duplicates it requires.

The input ensures that the data centers will always have enough machines for the new services.

Output

One line with N numbers: the remaining number of free machines of each data center, sorted in descending order.

Example

input	output

5 4
20 12 10 15 18
3 4
4 1
1 3
4 2
	11 10 10 9 8

Here are the detailed steps in the above example:

Time	Free machines	Operation
Beginning	20 12 10 15 18	
Service #1: before launching	20 18 15 12 10	sort the above line

Service #1: after launching	17 15 12 9 10	deduct 3 machines in 4 data centers
Service #2: before launching	17 15 12 10 9	sort the above line
Service #2: after launching	13 15 12 10 9	deduct 4 machines in 1 data center
Service #3: before launching	15 13 12 10 9	sort the above line
Service #3: after launching	14 12 11 10 9	deduct 1 machine in 3 data centers
Service #4: before launching	14 12 11 10 9	sort the above line
Service #4: after launching	10 8 11 10 9	deduct 4 machines in 2 data centers

Subtasks

subtask #	N≤	S≤
1	100	10
2	10 000	100
3	50 000	500

For all subtasks, the number of machines in any data center will not exceed 1 000 000.

Solutions

Subtask #1 & #2

Use some $O(N^2)$ sorting algorithm (Subtask #1) or O(N*log N) sorting algorithm (Subtask #2) to simulate the process.

Time complexity: $O(N^2*S)$ / O(N*log N*S)

Subtask #3

After launching each service, the data centers can be divided into two parts: one just deducted D machines each, another remains untouched. Each part remains its descending order. Therefore, we can merge these two parts like that in merge-sort, and it only takes O(N) time for each new service.

Time complexity: O(N*S)

Motivation of providing this problem

Sorting algorithms has great importance and is introduced in the very first chapters in *Introduction to Algorithms* and many other books.

However, some students only learn that the best time complexity for comparison sort is O(N*log N), and the “sort” in C++ library provides such functionality. They don’t really understand what’s behind all these sorting algorithms.

This problem tries to help the students thinking about the stories behind the basic sorting algorithms, and how to take advantage of each of them.