WOBURN CHALLENGE

2016-17 On-Site Finals

Saturday, May 6th, 2017

Senior Division Problems

Automated grading is available for these problems at:
wcipeg.com

For more problems from past contests, visit:
woburnchallenge.com

Problem S1: Cow-Bot Construction

9 Points / Time Limit: 3.00s / Memory Limit: 32M

It would seem that war is inevitable. Bo Vine's communication with his cow spy has been intercepted by the devious monkeys, confirming his suspicions that the Head Monkey is up to no good! In preparation for the impending conflict with the monkeys, Bo Vine has ordered the construction of a new, state-of-the-art Cow-Bot model from scratch.

The schematics call for N ($1 \le N \le 200{,}000$) different modules to be installed, in any order. Bo Vine's cow engineers require E ($1 \le E \le 10{,}000$) minutes to install any single module. However, the Cow-Bot (being a sophisticated, artificially intelligent piece of hardware) may also be able to help out! If at least M_i ($0 \le M_i \le N$) different modules have already been installed into the Cow-Bot up to a certain point in time, then it becomes able to install the i-th module into itself in B ($1 \le B \le 10{,}000$) minutes. Only one module may be in the process of being installed into the Cow-Bot at any time, meaning that the engineers and Cow-Bot may not simultaneously install two different modules at once.

Please help the cows determine the minimum amount of time required for all N modules to be installed into the Cow-Bot.

In test cases worth 7/9 of the points, $N \le 2000$.

Input Format

The first line of input consists of three space-separated integers N, E, and B.
N lines follow, the i-th of which consists of a single integer M_i (for $i = 1..N$).

Output Format

Output one line consisting of a single integer – the minimum number of minutes required for all N modules to be installed into the Cow-Bot.

Sample Input

```
7 7 4
4
0
4
2
6
4
4
```

Sample Output

```
34
```

Sample Explanation

One optimal sequence of module installations is as follows (with each step's completion time indicated):

- Module 2 by the Cow-Bot (4 minutes)
- Module 3 by the engineers (11 minutes)
- Module 7 by the engineers (18 minutes)
- Module 4 by the Cow-Bot (22 minutes)
- Module 6 by the Cow-Bot (26 minutes)
- Module 1 by the Cow-Bot (30 minutes)
- Module 5 by the Cow-Bot (34 minutes)

Problem S2: Rational Recipes

12 Points / Time Limit: 2.00s / Memory Limit: 16M

The worst has come to pass, with the war between the monkeys and cows now officially in full swing. In fact, the first major battle has been scheduled for next week! As an experienced military leader, the Head Monkey is well aware of the importance of an often under-appreciated aspect of war – the nutritional well-being of the soldiers. She's taken it upon herself to personally prepare the upcoming battle's rations.

The Head Monkey has N ($1 \le N \le 100$) different types of fruit ingredients to work with, numbered from 1 to N. Fruit type 1 corresponds to the monkeys' beloved bananas, of course, while the other fruit types are less tasty but have their own nutritional benefits. There are F_i ($1 \le F_i \le 1{,}000{,}000{,}000$) fruits of the i-th type available.

The Head Monkey intends to come up with a smoothie recipe, which will call for combining a certain set of fruits together to create a single smoothie. The recipe R will dictate that exactly R_i fruits of type i must go into the smoothie, where R_i is a strictly positive integer for each i between 1 and N. It's unclear exactly what recipe she'll come up with, though – we can only hope that it will be edible, for the monkeys' sake.

It's also unclear how many monkeys will actually be participating in the upcoming battle, though the number of monkeys will surely be some positive integer. However, it's imperative that each of them receive a single smoothie, with all of the smoothies having been created using the same recipe as one another.

Of course, the number of available fruits is a serious limiting factor. If there are m monkeys, and some recipe R is chosen, then $m \times R_i$ fruits of each type i will be required in total, and this quantity cannot exceed F_i. However, there's an additional restriction – due to the high value placed on bananas, it's important that there be no leftover bananas, as they'd go to waste. Therefore, it must be the case that $m \times R_1 = F_1$.

The Head Monkey has lots of great recipes in mind, but she's willing to accept that some of them might not work out in terms of producing a valid set of smoothies for 1 or more monkeys. That being said, she's wondering exactly how many different possible recipes she could validly choose. This quantity may be quite large, so she's only interested in its value when taken modulo 10,007.

As a hint, the following properties of modular arithmetic may be useful:

$(A + B) \bmod M = ((A \bmod M) + (B \bmod M)) \bmod M$
$(A \times B) \bmod M = ((A \bmod M) \times (B \bmod M)) \bmod M$

In test cases worth 2/12 of the points, $N \le 3$ and $F_i \le 100$ for each i.
In test cases worth another 3/12 of the points, $F_i \le 100$ for each i.
In test cases worth another 2/12 of the points, $F_i \le 10{,}000$ for each i.

Input Format

The first line of input consists of a single integer N.
The second line of input consists of N space-separated integers $F_1, \ldots, F_N$.

Output Format

Output one line consisting of a single integer – the number of different possible recipes which might be used (modulo 10,007).

Sample Input

```
3
4 2 4
```

Sample Output

```
10
```

Sample Explanation

3 possible recipes are as follows:

- R = [4, 1, 3] (serving 1 monkey)
- R = [4, 2, 1] (serving 1 monkey)
- R = [2, 1, 2] (serving 2 monkeys)

There are 7 other possible recipes.

Problem S3: Privacy

17 Points / Time Limit: 2.00s / Memory Limit: 64M

It's almost time to head into battle, but Bo Vine's cow soldiers insist on taking a nice long drink of water first. All N ($1 \le N \le 400$) of his army's cows have lined up at a trough to drink water. However, the cows like their privacy when drinking, and the i-th cow insists that they must drink from a trough from which at most C_i ($0 \le C_i < N$) other cows are also drinking.

The cows refuse to budge until they get hydrated, so to help make that possible, Bo Vine is prepared to install at most K ($0 \le K < N$) dividers at various points along the trough, effectively dividing it into multiple troughs as far as the cows are concerned. For example, if he installs a single divider between cows i and $i + 1$, then cows $1..i$ will be considered to drink from one trough, while cows $(i + 1)..N$ will be considered to drink from a different trough.

It may turn out that the cows' demands can't all be met even after the installation of K dividers. As such, Bo Vine may also need to "encourage" some of them to relax their requirements. Each cow is willing to increase their C value by 1 in exchange for 1 treat. Bo Vine may bribe any cow as many times as he'd like.

What's the minimum total number of treats which Bo Vine must give to the cows such that, once at most K dividers are installed, each cow will be willing to drink from its trough?

In test cases worth 4/17 of the points, $N \le 50$ and $K \le 1$.
In test cases worth another 8/17 of the points, $N \le 50$.

Input Format

The first line of input consists of two space-separated integers N and K.
N lines follow, the i-th of which consists of a single integer C_i (for $i = 1..N$).

Output Format

Output one line consisting of a single integer – the minimum total number of treats required such that the cows can all be satisfied after the installation of K dividers.

Sample Input

```
7 2
2 0 5 0 1 3 1
```

Sample Output

```
3
```

Sample Explanation

One optimal strategy is to give the second cow 2 treats to increase its C value from 0 to 2, and the fourth cow 1 treat to increase its C value from 0 to 1. Then, Bo Vine can install one divider between cows 3 and 4, and one more between cows 5 and 6, in order to yield the following valid set of troughs:

```
[ 2 2 5 | 1 1 | 3 1 ]
```

Problem S4: Bug Infestation

30 Points / Time Limit: 3.00s / Memory Limit: 64M

Having fortunately gotten wind of the Cow-Bot's construction before meeting it in battle, the monkeys are in the process of frantically creating a virus, with the hopes of installing it into the robot and shutting it down! They've finished writing their program, which consists of N ($1 \le N \le 300{,}000$) lines of code, but unfortunately its quality may be somewhat lacking...

At any point in time, each line of code either contains a bug, or doesn't. Initially, each line i contains a bug if $B_i = 1$, and otherwise doesn't if $B_i = 0$ ($0 \le B_i \le 1$).

Each minute, the monkeys can select one line of code which contains a bug, and fix that bug! Unfortunately, their code is so fragile that fixing some bugs may introduce others. If a bug is fixed on line i, then if $L_i = 0$, there are no consequences. Otherwise, one other line L_i ($1 \le L_i \le N$, $L_i \ne i$) will begin to have a bug (regardless of whether it already had one or not).

In order to efficiently proceed with making their viral code as correct as possible, the monkeys are interested in the answers to two questions. Firstly, what's the minumum possible number of lines of code which can be left containing bugs after they fix as many bugs as they'd like to? And secondly, how quickly can that minimum number of outstanding bugs be achieved? If $Q = 1$, then you only need to answer the first of these questions, and if $Q = 2$, then you must answer both.

In test cases worth 5/30 of the points, $Q = 1$ and $N \le 2000$.
In test cases worth another 4/30 of the points, $Q = 1$.
In test cases worth another 6/30 of the points, $N \le 2000$.

Input Format

The first line of input consists of two space-separated integers N and Q.
N lines follow, the i-th of which consists of two integers B_i and L_i (for $i = 1..N$).

Output Format

If $Q = 1$, then output a single integer – the minimum possible number of bugs which can remain in the code.
Otherwise if $Q = 2$, then output two space-separated integers – the minimum possible number of bugs which can remain in the code, and the minimum number of minutes required to achieve that number of bugs.

Sample Input

```
10 2
1 4
0 0
0 8
0 10
0 0
1 0
1 4
1 4
1 5
0 7
```

Sample Output

```
1 6
```

Sample Explanation

One optimal sequence of lines to debug is: 1, 6, 7, 8, 9, 5.
After this sequence, the only line containing a bug will be line 4. It's impossible to eliminate all of the bugs from the monkeys' code (hopefully the same isn't true for yours…).

Problem S5: Bovine Grenadiers

32 Points / Time Limit: 4.00s / Memory Limit: 64M

Angus and Bessie are the most well-known grenadiers in Bo Vine's army, and they're constantly trying to out-do one another. Even on the very eve of the war's first major battle, they've ended up in an argument! On this occasion, they're fighting over how to split up the army's supply of grenades - of course, each of them wants the most powerful grenades to themselves! In an attempt at fairness, they're going to play a game to divide up the grenades.

There are N ($1 \le N \le 300{,}000$) boxes of grenades. The i-th box contains G_i ($1 \le G_i \le 300{,}000$) grenades, with the j-th of those grenades having a "grenade power" of $P_{i,j}$ ($1 \le P_{i,j} \le 10{,}000{,}000$), indicating its explosive strength. There are at most 300,000 grenades in total across all of the boxes.

Angus and Bessie will take turns performing actions, with Angus going first, until each of the grenades has been taken by one of them. All N of the boxes are initially sealed. In one turn, a cow may either unseal a sealed box, or take one grenade from an unsealed box. Both cows will make optimal actions with the goal of maximizing the total grenade power of the grenades that they'll get their hands on throughout the game (and thus minimizing the total grenade power obtained by their opponent).

To make things more exciting, the entire game will actually be independently re-played M ($1 \le M \le 300{,}000$) separate times. Before the i-th time the game gets played, a single grenade will get swapped out for a slightly different one. In particular, the B_i-th grenade in the A_i-th box ($1 \le A_i \le N$, $1 \le B_i \le G_{A_i}$) will be replaced with a new grenade whose grenade power is D_i ($-1 \le D_i \le 1$) larger than that of the removed grenade. It's guaranteed that the new grenade's power will still be positive. Each such replacement will carry over to all subsequent times the game gets played, and the new grenade itself may get replaced later on.

Help Angus and Bessie determine the outcome of their set of games by predicting how much grenade power each of them will end up with each time they play. Rather than outputting all $2M$ such values (the grenade power obtained by Angus and Bessie each time they play), you only need to compute the sum of Angus's M values, as well as the sum of Bessie's M values.

In test cases worth 5/32 of the points, $N = 1$, $M \le 2000$, and $G_1 \le 2000$.
In test cases worth another 4/32 of the points, $N = 1$.
In test cases worth another 6/32 of the points, $N \le 2000$, $M \le 2000$, and there are at most 2000 grenades in total.

Input Format

The first line of input consists of two space-separated integers N and M.
N lines follow, the i-th of which consists of an integer G_i, followed by a space, followed by G_i space-separated integers $P_{i,1}$, …, P_{i,G_i} (for $i = 1..N$).
M lines follow, the i-th of which consists of three space-separated integers A_i, B_i, and D_i (for $i = 1..M$).

Output Format

Output a single line consisting of two space-separated integers – the total grenade power obtained by Angus and Bessie, respectively.

Sample Input

```
3 3
2 4 3
1 5
2 1 1
3 2 1
3 2 1
2 1 -1
```

Sample Output

```
17 29
```

Sample Explanation

After the first grenade replacement, assuming both cows play optimally, Angus will end up obtaining 5 grenade power (for example, he may get 4 from the 1st grenade in the 1st box, and 1 from the 1st grenade in the 3rd box), while Bessie will obtain 10 from the remaining grenades.
After the second replacement, Angus will obtain 6 while Bessie will obtain 10. After the third replacement, Angus will obtain 6 while Bessie will obtain 9. In total, then, Angus will have obtained 5 + 6 + 6 = 17 grenade power, while Bessie will have obtained 10 + 10 + 9 = 29.