PROBLEM 5

Tennis Robot II

Input file: `tennisin.txt`
Output file: `tennisout.txt`

Time and memory limits: 1 second, 1 GB

The cleaning robot at your local sports centre has been hard at work since they bought it several years ago. However, years of reprogramming has left its code a complete mess!

Currently, the sports centre has N bins used to store tennis balls. The ith bin starts with X_i tennis balls but has enough space to store as many extra balls as necessary.

The cleaning robot is currently programmed with M instructions. The jth instruction is to take one tennis ball from bin A_j and move it to bin B_j. The robot will carry out the instructions in order, and once it completes the final instruction it will loop back to the 1st one and continue. It will only stop if it is instructed to take a ball from an empty bin, at which point it will crash and shut down.

You want to know how long it will take for the robot to stop, so that you can fix it to do something more useful (it's not even cleaning, it's just moving balls between bins!).

How many instructions will the robot successfully complete before stopping, or will it run forever?

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

- $2 \leq N \leq 200\,000$.
- $1 \leq M \leq 200\,000$.
- $1 \leq X_i \leq 1\,000\,000$ for all i.
- $1 \leq A_j, B_j \leq N$ and $A_j \neq B_j$ for all j.

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (25 marks), $N, M \leq 1000$ and $X_i \leq 1000$ for all i.
- For Subtask 2 (40 marks), $A_j \neq A_k$ for all $j \neq k$. In other words, all instructions take tennis balls from different bins[1].
- For Subtask 3 (35 marks), no special rules apply.

Input

Your program must read input from the file `tennisin.txt`. When testing on your own computer, this file must be placed in the same folder as your program. We strongly recommend using the solution templates (which you can find on the *Templates & Downloads* page of the competition website) to help you with input and output.

The file `tennisin.txt` follows a specific format:

- The 1st line of input contains the integers N and M.
- The 2nd line of input contains N integers describing the number of balls that start in each bin. The ith of these is X_i.
- The next M lines describe the robot's instructions. The jth of these lines contains the two integers A_j and B_j.

[1] Sample input 1 satisfies the rules of subtask 2, but sample inputs 2 and 3 do not.

Output

Your program must write to the file `tennisout.txt`. If the robot will run forever your program must output `FOREVER`. Otherwise, it must output the number of instructions the robot will successfully complete before stopping.

Note: For students using C, C++ or Java, please note that the answer may exceed the maximum value that can be stored in an `int` *integer type. As such, you should consider using the* `long long` *integer type in C or C++, or the* `long` *integer type in Java. Please refer to the solution templates (which you can find on the* Templates & Downloads *page of the competition website) for more details. Python users do not need to consider this.*

Sample input 1

```
4 3
2 1 1 3
1 2
3 1
2 4
```

Sample input 2

```
3 4
3 2 4
2 3
3 2
1 2
2 1
```

Sample input 3

```
5 4
3 2 6 4 5
1 5
4 3
2 3
4 2
```

Sample output 1

```
4
```

Sample output 2

```
FOREVER
```

Sample output 3

```
9
```

Explanation

- In the 1st sample case, there are 4 bins and they start with 2, 1, 1, and 3 tennis balls. Here are the instructions that the robot executes and the number of tennis balls in the bins after each instruction.

Instruction	Successful?	Bin 1	Bin 2	Bin 3	Bin 4
Bin 1 to bin 2	Yes	1	2	1	3
Bin 3 to bin 1	Yes	2	2	0	3
Bin 2 to bin 4	Yes	2	1	0	4
Bin 1 to bin 2	Yes	1	2	0	4
Bin 3 to bin 1	No	–	–	–	–

  The robot successfully executes 4 instructions, then crashes on the next instruction when it tries to take a ball from bin 3, which is empty.

- In the 2nd sample case, the robot will continue executing instructions forever.
- In the 3rd sample case, the robot will successfully execute 9 instructions before crashing.