

Por Una Cabeza

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

Bobo is the director of the famous TV Show “Sunday Night Live”. The show recently introduced a new stage named “World Debate”, which has been widely acclaimed since its first appearance. Despite its possibly daunting name, it is a stage where the audience votes on some particular topic. Specifically, the show staff will choose a controversial topic, and the audience will vote 0 (meaning “not agree”) or 1 (meaning “agree”) towards it. These votes will then be collected to produce a final opinion on this topic.

What makes this stage particularly stand out from the rest is the use of *voting machines*. Voting machines are automatic systems that receive an odd number of inputs in $\{0,1\}$ and take the **majority** of input received as the output. **Note that it is not necessary for the voting machine to receive all inputs to produce an output.** Once a voting machine receives an amount of a same value that is more than half of its input slots, it will immediately produce the value as an output.

The show staff will then use these voting machines to produce a result. Suppose there are n audience numbered from 1 to n , and m voting machines numbered from 1 to m . The final opinion will be produced in the following manner:

- **All audience votes in sequential order from 1 to n .**
- Each voting machine takes its input from either some audience’s vote or the output of a voting machine **with a number smaller than it.**
- The vote of each audience and the output of each voting machine except the one numbered m is taken as the input of **exactly one** voting machine.
- The output of the voting machine numbered m will be taken as the final opinion.

As the director, Bobo does not really care about the final result. All he cares about is the rating of the show. He wants to create an intense state called “Por Una Cabeza” so that **for every voting machine, it doesn’t produce an output until it receives all its inputs.** This may not be possible if all audience votes by their own will, but Bobo can make a difference. Bobo knows the vote a_i of the audience numbered i ($1 \leq i \leq n$), and the cost b_i Bobo needs to pay to make the audience change his/her vote to the other side. Under a quite tight budget, Bobo wants to know the minimum cost to achieve “Por Una Cabeza”.

But wait! The TV show is live, so situations might change. The a_i and b_i may undergo a total of q **permanent** changes, and Bobo needs to know the minimum cost to achieve “Por Una Cabeza” after each change. Bobo has put too much effort into this TV show and is too tired. Can you help him?

Input

The first line contains two integers n, m, q ($1 \leq n, m, q \leq 10^5$), denoting the number of audiences, voting machines, and changes, respectively.

The n lines follow. The i -th ($1 \leq i \leq n$) line contains two integers a_i, b_i ($a_i \in \{0,1\}, 1 \leq b_i \leq 10^9$), denoting the votes and the cost needed to change the vote of the i -th audience.

Then the description of the input of m voting machines follows, with order sequentially from 1 to m . The input of each voting machine is described by a line of the form $\ell, c_1, c_2, \dots, c_\ell$ ($1 \leq \ell \leq n + m - 1, -m \leq c_i \leq n, c_i \neq 0, c_i$ is pairwise distinct, ℓ is odd), where $c_i > 0$ means the vote of audience numbered c_i is taken as an input of the current voting machine, and $c_i < 0$ means the output of voting machine numbered $-c_i$ is taken as an input of the current voting machine.

The q lines describing the changes follow. Each line is of the form x, y, z ($1 \leq x \leq n, y \in \{0,1\}, 1 \leq z \leq 10^9$), meaning a_x will be changed to y and b_x will be changed z . The change is **permanent**, meaning it will not be immediately rolled back after the query.

It is guaranteed each voting machine takes its input from either some audience’s vote or the output of a voting machine **with a number smaller than it**.

It is guaranteed that the vote of each audience and the output of each voting machine except the one numbered m is taken as the input of **exactly one** voting machine.

Output

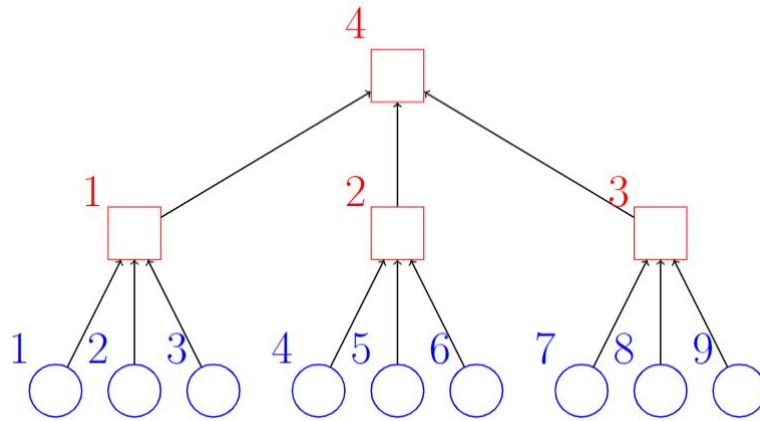
After each of the q changes, output a number in a line, denoting the minimum cost for Bobo to achieve “Por Una Cabeza”. It can be proven it is always possible for Bobo to achieve “Por Una Cabeza” under the given input constraints.

Examples

standard input	standard output
3 1 5	2
0 2	2
1 3	0
0 2	1
3 1 2 3	1
2 0 3	
2 0 3	
2 1 3	
1 1 1	
3 1 1	
9 4 9	0
0 1	6
1 2	3
0 3	6
1 4	10
0 5	6
1 6	3
0 7	4
1 8	3
0 9	
3 1 3 2	
3 6 4 5	
3 9 8 7	
3 -1 -2 -3	
9 1 4	
8 0 6	
7 0 3	
6 0 8	
5 1 5	
4 0 7	
3 1 2	
2 0 9	
1 1 1	

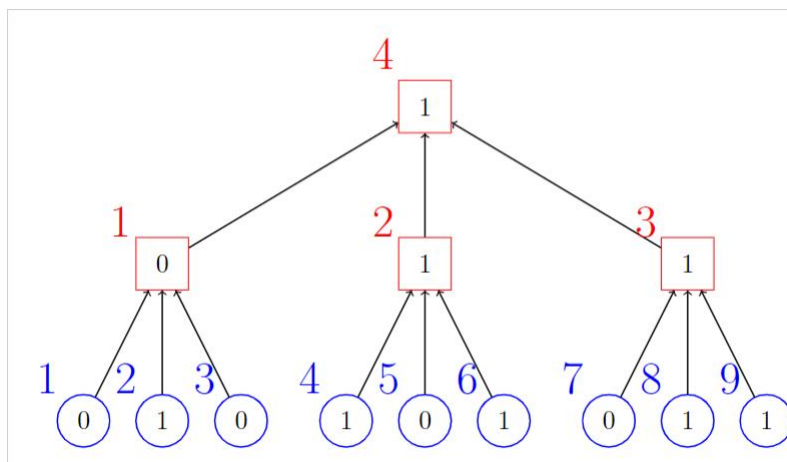
Note

Here is an example of the structure formed by the audience and voting machines in the second sample test, where circles represent the audience and the squares represent the voting machines, both labeled with corresponding numbers, and the output from the start of an arrow is given as the input of the end of the arrow. Note that the output of machine number 4 will be produced as the final result.



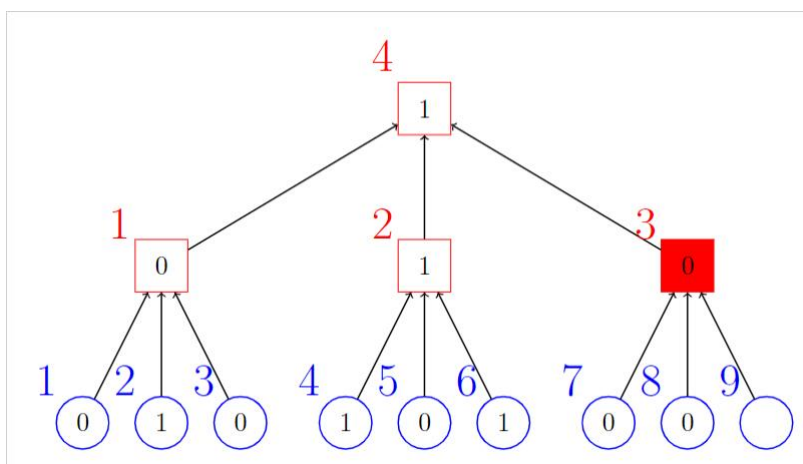
the structure in the second sample test

After the first change, the result is given in the following picture. Here for every voting machine, it doesn't produce an output until it receives all its inputs. So "Por Una Cabeza" is already achieved and there's no need to change the vote of any person and minimum cost is therefore, 0.



the second sample test, after the first change

After the second change, the result is given in the following picture. Here after the 8-th person voted, the outcome of the third voting machine is already decided, without receiving all its inputs, so "Por Una Cabeza" is not achieved and it is optimal for Bobo to change the vote of the 8-th person, which costs 6 exactly after this change.



the second sample test, after the second change