
Legs warm-up exercise

Input file: **standard input**
Output file: **standard output**
Time limit: 6 seconds
Memory limit: 512 megabytes

Once upon a time, a boy Kirill tried hard to solve a problem with the intriguing name «Hands warm-up exercise». Whether or not he managed to solve the problem, we do not know. However, during his contemplation, Kirill came up with a new problem for you to solve. The condition of the problem is as follows:

In the country X , there are n cities connected by m roads. The set of roads has two interesting properties. Firstly, all roads are one-way. Secondly, even if the roads were bidirectional, there would be no more than one simple path between any pair of cities. It is easy to notice that without the orientation, the graph with the given n cities as vertices and m roads as edges is a forest (meaning that in each connected component, the number of edges is one less than the number of vertices).

The function $cost(u, v)$ is defined as follows: if there exists a path from u to v , then it is equal to the number of roads on that path, and if there is no path, then it is equal to 0.

The *beauty* of the country map is defined as $\sum_{u=1}^n \sum_{v=1}^n cost(u, v)$, which is the sum of the values of $cost$ for all pairs of cities.

The government of country X loves to calculate various statistics. This time, it was decided to calculate the *beauty* of the current country map, as well as process q changes to the country map, recalculating its *beauty* after each change.

There are 4 types of changes:

1. Add a road from city u to city v .
2. Remove the road between cities u and v .
3. Reverse the orientation of the road between cities u and v .
4. Orient all roads on the path from u to v in the direction from u to v . It is guaranteed that this is possible, i.e. there exists a path from city u to city v along the roads, ignoring their orientation.

It is guaranteed that after each change, the set of cities and roads continues to form a forest.

Note that in this problem, the queries modify the country map and depend on previous modifications.

Input

The first line contains an integer g ($0 \leq g \leq 10$) — the test group.

The second line contains three integers n , m and q ($2 \leq n \leq 400\,000$, $0 \leq m < n$, $1 \leq q \leq 400\,000$) — the number of cities, roads and changes, respectively.

Each of the next m lines contains two integers u and v ($1 \leq u, v \leq n$) — the roads on the initial map oriented from u to v .

Each of the next q lines contains three integers t , u , and v ($1 \leq t \leq 4$, $1 \leq u, v \leq n$) — the type of query and the pair of vertices that define it.

Output

In the first line, print the initial *beauty* of the country map.

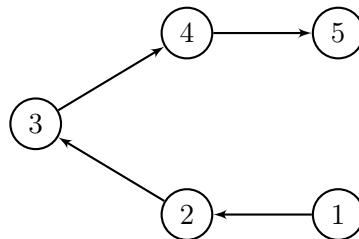
In the next q lines, print one number per line — the values of the *beauty* of the country map after each change.

Example

standard input	standard output
0	20
5 4 4	6
1 2	3
2 3	4
3 4	16
4 5	
3 4 3	
2 3 2	
1 4 2	
4 1 4	

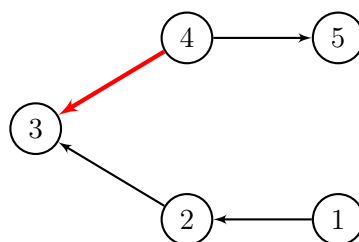
Note

In the test example, before any changes, the graph looks like this:

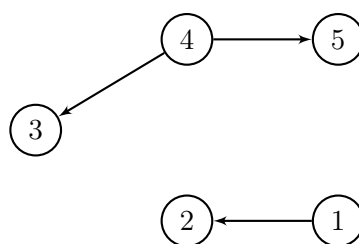


In this case, for example, $cost(1,2) = cost(2,3) = cost(3,4) = cost(4,5) = 1$, because there is a direct road between these pairs of cities. Similarly, $cost(1,3) = cost(2,4) = cost(3,5) = 2$, since the second city in each pair can only be reached in two roads from the first. Similarly, for example, $cost(2,1) = cost(4,1) = cost(5,3) = 0$, because the second city in the pair is not reachable from the first. Summing these values for all pairs of cities, we get 20.

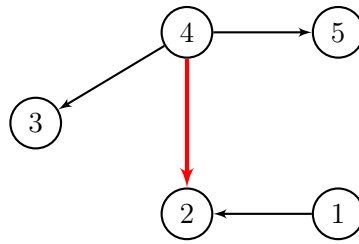
After that, we need to change the direction of the road between cities 3 and 4. Since it was previously from city 3 to city 4, now it should go from city 4 to city 3. After this, the answer to the problem is 6 and the graph looks like this:



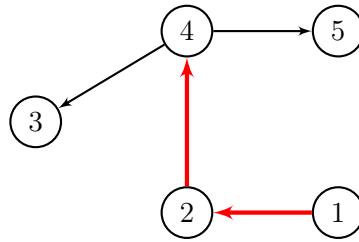
Next, the road between cities 2 and 3 is removed. After this, the answer to the problem is 3, and the graph looks like this:



After that, a road from city 4 to city 2 is added. After this, the answer to the problem is 4, and the graph looks like this:



After that, all roads between cities 1 and 4 are oriented from city 1 to city 4. After this, the answer to the problem is 16, and the graph looks like this:



Scoring

The tests for this problem consist of ten groups. Points for each group are awarded only if all the tests in that group and some tests from the previous groups pass. **Offline-testing** means that the results of testing your solution on this group will only be available after the competition ends.

Group	Score	Additional constraints		Required groups	Comment
		n, m, q	Changes types		
0	0	–	–	–	Samples.
1	10	$n, m, q \leq 100$	–	0	
2	8	$n, m, q \leq 5000$	–	0, 1	
3	11	$n, m, q \leq 100\,000$	1	–	
4	7	$n, m, q \leq 100\,000$	1, 2	3	
5	13	$n, m, q \leq 100\,000$	3	–	
6	9	$n, m, q \leq 100\,000$	3, 4	5	
7	12	$n, m, q \leq 100\,000$	–	–	$ u - v = 1$ always, except for changes of type 4.
8	18	$n, m, q \leq 100\,000$	–	0 – 7	
9	6	$n, m, q \leq 200\,000$	–	0 – 8	Offline-testing.
10	6	–	–	0 – 9	Offline-testing.