G. Great Cactus Comeback

Problem ID: cactus

Did you know that cactus graphs once enjoyed a glorious era in Taiwan's competitive programming scene, especially in 2022?

Looking back, the 2021 NTU ICPC Team Preliminary Contest featured a problem that required answering multiple shortest-path distance queries on a cactus graph[1], marking the beginning of the Great Cactus Era. In 2022, cactus graphs reached the height of their glory. First, IOICamp featured a problem about answering maximum contiguous path-sum queries between pairs of vertices in a cactus graph[2]. Later, YTP featured a problem asking for the maximum-weight independent set of a cactus graph[3]. Incidentally, this problem was so difficult that even the first-place team solved every problem except this one! Finally, that year's NTU ICPC Team Preliminary Contest featured yet another problem combining cactus graphs with linear basis[4].

Unfortunately, since 2023, cactus graphs have almost completely disappeared from programming contests in Taiwan. What a shame! Now, you are given a simple, connected, undirected cactus graph with n vertices and m edges. Each edge has a positive integer weight. There is a set of vertices S which is initially empty. Please process q operations. Each operation specifies a vertex v and toggles whether v belongs to S. That is,

- If $v \in S$, remove v from S;
- If $v \notin S$, add v to S.

After each operation, output the maximum shortest-path distance between any two vertices in S. That is, let $d(u, v)$ be the length of the shortest path between vertices u and v. You should output

$$\max_{u,v \in S} d(u, v)$$

after each operation. Note that if $|S| \leq 1$, the answer is defined to be 0.

To make the problem more interesting, all operations are encoded and must be processed online. See the input format for details.

Note that an undirected graph is called a cactus graph if every edge belongs to at most one simple cycle.

Input

The first line contains three integers n, m, and q, denoting the number of vertices, the number of edges, and the number of operations, respectively.

Each of the next m lines contains three positive integers u_i, v_i, and w_i, indicating that there is an undirected edge of weight w_i between vertices u_i and v_i.

Each of the next q lines contains an integer x_i, representing the encoded information for the i-th operation.

Let ans_i be the answer after the i-th operation, and let y_i be the vertex that is actually toggled by the i-th operation. Then

$$y_i = \text{ans}_{i-1} \oplus x_i,$$

where $\oplus$ denotes the bitwise XOR operation. Also, we define $\text{ans}_0 = 0$.

- $1 \le n, q \le 10^5$
- $0 \le m \le 2 \times 10^5$
- $1 \le u_i, v_i \le n$
- $1 \le w_i \le 10^9$
- $0 \le x_i < 2^{60}$
- $1 \le y_i \le n$
- It is guaranteed that the input graph is a simple, connected cactus graph.

Output

Output q lines. The i-th line should contain one integer ans_i, the maximum shortest-path distance between any two vertices in S after the i-th operation.

Sample Input 1	Sample Output 1
15 19 20 10 13 4 11 1 2 14 12 5 9 5 5 8 5 9 14 8 10 15 11 5 13 3 3 6 3 6 13 6 10 13 7 1 1 15 5 4 15 8 7 10 8 14 2 6 14 10 5 12 8 1 4 3 7 2 10 7 6 14 31 27 36 44 37 42 47 17 30 44 18 13 0 41 6 8 31 30	0 18 18 38 38 38 38 38 23 17 32 29 14 11 34 11 7 29 27 42

Sample Input 2	Sample Output 2
8 8 10 2 1 5 2 7 3 8 5 4 5 4 1 6 4 5 3 2 4 1 6 5 4 2 2 2 8 2 6 13 4 9 15 10 4	0 7 7 12 7 11 11 11 12 9

Note

In the first sample, the actual query are: 2 8 9 12 3 5 13 6 6 9 6 6 15 14 15 4 4 5 10 1.

In the second sample, the actual query are: 2 8 5 1 1 3 2 4 1 8.