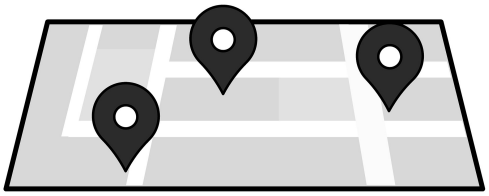
# Problem K
## Kleptocrat



Your company has a policy that every employee should be refunded an amount of money proportional to the shortest distance between their home and office. This causes the loophole that many employees intentionally move very far away to claim the maximum possible reimbursement.

One employee has taken this policy way too far and is in danger of bankrupting you. You must find a way to stop this before cancelling the policy next year. However, the rules are strict: as long as the employee keeps track of the distances they have travelled, you are forced to reimburse them.

Suddenly you have a flash of inspiration: nowhere does it say that you have to use the *Euclidean* distances! You start working on more subtle distance functions and now you have a first prototype: XOR distance. The length of a path is defined as the XOR of the lengths of the edges on the path (as opposed to the sum). The distance between two locations is defined as the length of the shortest path between them.

You will need to test this principle on the transport network with the locations of each of your employees in turn.

## Input

- One line containing three integers $n$ ($2 \leq n \leq 10^4$), $m$ ($n - 1 \leq m \leq 10^5$), and $q$ ($1 \leq q \leq 10^5$), the number of nodes, edges, and questions respectively.
- $m$ lines describing an edge. Each line consists of three integers $x$, $y$, $w$ ($1 \leq x, y \leq n$, $x \neq y$ and $0 \leq w \leq 10^{18}$), indicating that there is an undirected edge of length $w$ between nodes $x$ and $y$.
- $q$ lines describing a question. Each line consists of two integers $a$, $b$ ($1 \leq a, b \leq n$) asking for the shortest distance between nodes $a$ and $b$.

Between every pair of distinct nodes, there is at most one edge, and every node can be reached from any other node.

## Output

For every question, output one line containing the shortest distance between nodes $a$ and $b$.

**Sample Input 1**

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

**Sample Output 1**

```
1
1
0
```

**Sample Input 2**

```
7 10 5
1 2 45
2 3 11
2 4 46
3 4 28
3 5 59
3 6 12
3 7 3
4 5 11
5 6 23
6 7 20
1 4
2 6
3 5
1 7
5 5
```

**Sample Output 2**

```
1
5
0
5
0
```