Problem A
Awesome MST Problem

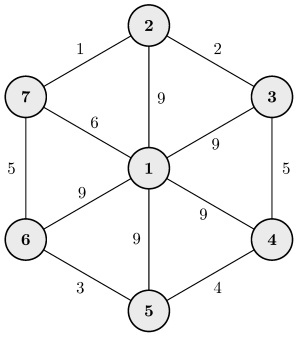
Given a connected, weighted undirected graph with N vertices and M edges. The graph doesn't have any self-loops, and between any pair of vertices, there is at most one edge.

An edge is called **awesome** if there is **at least one** minimum spanning tree containing that edge.

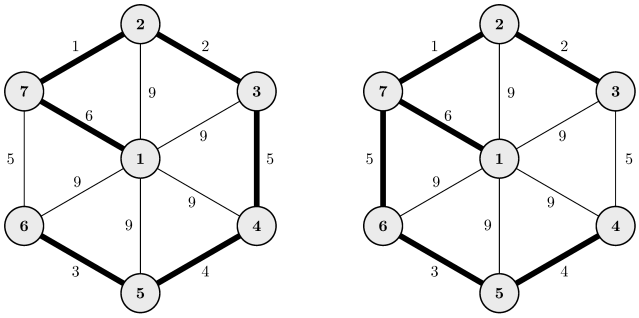
You are given Q queries. In each query, you need to add one new edge to the given graph, and then count the number of **awesome** edges in the new graph. Note that once an edge is added, it stays in the graph forever.

A **minimum spanning tree** of a connected, weighted undirected graph is a subset of its edges that connects all its vertices together, without any cycles and with the minimum possible total edge weight.

For example, consider the following graph:



It has 2 **minimum spanning trees**, presented in two figures below, whose edges are in bold:



The graph has 7 **awesome** edges, which appear in at least one minimum spanning trees: $(2, 7)$, $(2, 3)$, $(7, 1)$, $(7, 6)$, $(3, 4)$, $(6, 5)$ and $(4, 5)$.

Input

The first line of the input contains two integers N and M $(1 \le N, M \le 10^5)$ — the number of vertices and edges of the graph, respectively.

In the next M lines, the i^{th} one contains three integers: u_i, v_i, and w_i ($1 \le u_i, v_i \le N; 0 \le w_i \le 10^9; u_i \ne v_i$), indicating that there is an edge connecting two vertices u_i and v_i of weight w_i in the initial graph.

The next line contains a single integer Q ($1 \le Q \le 10^5$) — the number of queries.

In the next Q lines, each contains three integers x, y and z ($1 \le x, y \le N, 0 \le z \le 10^9, x \ne y$) describing a query in which you need to add an edge of weight z connecting two vertices x and y.

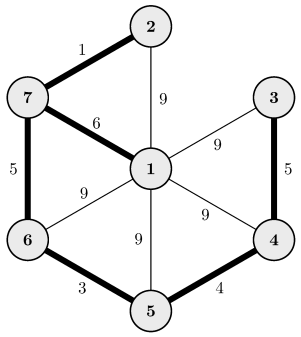
It is guaranteed that at the beginning and after every query, the graph is always connected and no pairs of vertices are connected by more than one edges.

Output

For each query, print a single line containing the number of **awesome** edges in the graph after that query.

Explanation of the sample input

After the first query, the graph is as below:



After the second query, the graph is the same as the graph in the problem description.

Sample Input 1	Sample Output 1
7 10 2 7 1 5 6 3 4 5 4 6 7 5 3 4 5 1 7 6 1 2 9 1 3 9 1 4 9 1 5 9 2 1 6 9 2 3 2	6 7