Problem F. Finite Walking

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are given an undirected graph G with n vertices and m edges. Loops and multiple edges are allowed. Each edge has a positive integer a_i assigned to it, and a counter b_i which is initially zero.

Consider the following process on this graph: we choose the starting vertex v_1 arbitrarily, then we go along some edge e_1 starting in v_1 and ending in v_2, then we go along some edge e_2 from v_2 to v_3 and so on. In other words, we follow some path in the graph, which may contain some edges and vertices multiple times. Every time we traverse edge i in any direction, its counter b_i changes to $(b_i + 1) \bmod a_i$. We can stop the process after any number of steps (possibly, zero).

Let us call the array $(b_1, b_2, \ldots, b_m)$ after this process a *configuration array*. Consider all possible processes corresponding to all possible finite paths in G. How many different configuration arrays can they produce? Two configuration arrays are considered different if they differ in at least one element. As the answer can be quite large, give it modulo $10^9 + 7$.

Input

The first line of input contains two space-separated integers n, m ($1 \le n \le 2 \cdot 10^5$, $0 \le m \le 4 \cdot 10^5$) — number of vertices and number of edges of the graph respectively.

Next m lines contain the description of edges, one per line: i-th edge is described by three integers u_i, v_i, a_i ($1 \le u_i, v_i \le n$, $1 \le a_i \le 10^9$) — the endpoints of the edge and the number assigned to this edge. Vertices are indexed starting from 1.

Note that loops and multiple edges are **allowed**.

Output

In the only line print the number of different configuration arrays modulo $10^9 + 7$.

Example

standard input	standard output
5 5 1 2 2 2 3 2 3 4 2 4 5 2 5 1 1	14