The Quest for El Dorado

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

There is a kingdom with n cities and m bi-directional railways connecting the cities. The i-th railway is operated by the c_i-th railway company, and the length of the railway is l_i.

You'd like to travel around the country starting from city 1. For your travel, you have bought k railway tickets. The i-th ticket can be represented by two integers a_i and b_i, meaning that if you use this ticket, you can travel through some railways in one go, as long as they're all operated by company a_i and their total length does not exceed b_i. It is also allowed to just stay in the current city when using a ticket. You can only use the tickets one at a time, and you can only use each ticket once.

As you find it a burden to determine the order to use the tickets, you decided to use the tickets just in their current order. More formally, you're going to perform k operations. In the i-th operation, you can either choose to stay in the current city u; Or choose a different city v, such that there exists a path from city u to city v where all railways in the path are operated by company a_i and the total length of railways does not exceed b_i, and finally move to city v.

For each city, determine if it is possible to arrive in that city after using all k tickets.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains three integers n, m, and k ($2 \le n \le 5 \times 10^5$, $1 \le m \le 5 \times 10^5$, $1 \le k \le 5 \times 10^5$) indicating the number of cities, the number of railways and the number of tickets.

For the following m lines, the i-th line contains four integers u_i, v_i, c_i, and l_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $1 \le c_i \le m$, $1 \le l_i \le 10^9$) indicating that the i-th railway connects city u_i and v_i. It is operated by company c_i and its length is l_i. Note that there might be multiple railways connecting the same pair of cities.

For the following k lines, the i-th line contains two integers a_i and b_i ($1 \le a_i \le m$, $1 \le b_i \le 10^9$) indicating that if you use the i-th ticket, you can travel through some railways in one go, if they're all operated by company a_i and their total length does not exceed b_i.

It's guaranteed that the sum of n, the sum of m, and the sum of k of all test cases will not exceed 5×10^5.

Output

For each test case, output one line containing one string $s_1 s_2 \cdots s_n$ of length n where each character is either 0 or 1. If you can travel from city 1 to city i with these k tickets, then $s_i = 1$; Otherwise $s_i = 0$.

Example

standard input	standard output

2
5 6 4
1 2 1 30
2 3 1 50
2 5 5 50
3 4 6 10
2 4 5 30
2 5 1 40
1 70
6 100
5 40
1 30
3 1 1
2 3 1 10
1 100
	11011
	100

Note

For the first sample test case:

- To arrive in city 4, you can use the 1-st ticket to move from city 1 to city 2, then stay in city 2 when using the 2-nd ticket, then use the 3-rd ticket to move from city 2 to city 4, then stay in city 4 when using the 4-th ticket.
- To arrive in city 5, you can use the 1-st ticket to move from city 1 to city 5 by going through the 1-st and the 6-th railway, then stay in city 5 when using the following tickets.
- As you cannot change the order to use the tickets, you cannot arrive in city 3.