Problem M: Marathon Tournament

Time limit: 3s; Memory limit: 512 MB

This year, an ICPC Regional Contest is going to be hosted in Hanoi. Together with a very tough contest to determine world finalists, the organizers decide to host a marathon event, giving contestants opportunities to explore the loveliness of Hanoi.

The venue of this marathon event consists of n junctions. These junctions are numbered from 1 to n. There are m bidirectional roads, which are numbered from 1 to m. The i-th one connects two junctions u_i and v_i and has a length of l_i. These roads guarantee that travelling between every pair of junctions is always possible.

The marathon tournament consists of multiple rounds. In each round, participants start at some junction, pass through several roads, and finish at some other junction. All participants run on exactly the same route, which is chosen by the organizers. The following conditions hold for all rounds:

- There is going to be at least one rounds.
- In all rounds, the starting junction differs from the finishing one.
- The starting junction of every round is the same as the finishing junction of the previous round.
- In all rounds, the organizers always choose a shortest path from the starting junction to the finishing junction. A shortest path is a path with minimum possible total length of roads

The organizers define the *difficulty set* of the tournament as the set of distinct values of rounds' lengths. Formally, a tournament consisting of k rounds can be represented as a sequence of junctions $x_0, x_1, \ldots, x_k$ so that:

- The first round starts at junction x0 and finishes at junction x_1,
- The second round starts at junction x1 and finishes at junction x_2,
- The third round starts at junction x2 and finishes at junction x_3,
- … ,
- The last round starts at junction xk-1 and finishes at junction x_k.

Let $d(a, b)$ be the distance of a shortest path from junction a to junction b. The *difficulty set* of the above tournament is the set of values which appears in the following sequence at least once: $d(x_0, x_1)$, $d(x_1, x_2)$,…, $d(x_{k-1}, x_k)$.
To make an interesting tournament, the organizers have q plans of choosing routes. In each plan, the organizers have a set of numbers $s_1, s_2, \ldots, s_c$. The organizers would like to know if it is possible to make a tournament so that its *difficulty set* equals to this set.

Please help the organizers for a wonderful ICPC event!

Input

The first line of the input contains a single integer θ – the number of test cases. Then θ test cases follow, each is presented as below:

- The first line is an empty line.
- The second line contains three integers n, m and q ($1 \le n \le 400$, $0 \le m \le n(n-1)/2$, $1 \le q \le 2\times10^5$) – the number of junctions, the number of roads and the number of plans, respectively.
- In the next m lines, the i-th one contains three integers u_i, v_i and l_i ($1 \le u_i, v_i \le n$; $1 \le l_i \le 10^4$) representing the i-th road.
- In the last q lines, each contains an integer c ($1 \le c \le 2\times10^5$) followed by c integers $s_1, s_2, \ldots, s_c$ ($1 \le s_1 < s_2 < \ldots < s_c \le 10^7$) representing a plan.

It is guaranteed that:

- The sum of n over all test cases does not exceed 1600.
- The sum of c over all plans over all test cases does not exceed 8×10^5.

Output

- For each test case, print a single line containing exactly q words. The i-th one among them represents the answer for the i-th plan, which is either **Yes** (meaning that it is possible to make such a tournament) or **No** (no such tournament exists).

Sample

Input	Output
2 5 4 3 3 4 1 2 1 2 3 2 3 4 5 4 2 1 2 2 3 4 3 3 4 8 5 4 3 3 4 1 2 1 2 3 2 3 4 5 4 2 4 6 2 3 5 3 4 5 6	No Yes Yes Yes Yes Yes

Explanation:

Explanation first test case:

- In the second plan, one possible tournament is: 3 → 2 → 4. We have $d(3, 2) = 3$ and $d(2, 4) = 4$.
- In the third plan, one possible tournament is: 3 → 2 → 4 → 5 → 2. We have $d(3, 2) = 3$, $d(2, 4) = d(4, 5) = 4$ and $d(5, 2) = 8$.