You are the mayor of a city with severe traffic problems. To deal with the situation, you have decided to make a new plan for the street grid. As it is impossible to make the streets wider, your approach is to make them one-way (only traffic in one direction is allowed on a street), thus creating a more efficient flow of traffic.

The streets in the city form an orthogonal grid — like on Manhattan avenues run in north-south-direction, while streets run in east-west-direction. Your mission is to make all the streets and avenues one-way, i.e. fix the direction in which traffic is allowed, while maintaining a short driving distance between some ordered pairs of locations. More specifically, a route in the city is defined by two street-avenue crossings, the start and goal location. On a one-way street grid, a route has a legal path if it is possible to drive from the start location to the goal location along the path passing streets and avenues in their prescribed direction only. A route does not define a specific path between the two locations - there may be many possible paths for each route. A legal path in a one-way street grid is considered simple if it requires at most one turn, i.e. a maximum of one street and one avenue need to be used for the path.

When traveling by car from one location to another, a simple path will be preferred over a non-simple one, since it is faster. However, as each street in the grid is one-way, there may always be routes for which no simple path exists. On your desk lies a list of important routes which you want to have simple paths after the re-design of the street grid.

Your task is to write a program that determines if it is possible to fix the directions of the one-way streets and avenues in such a way that each route in the list has at least one simple path.

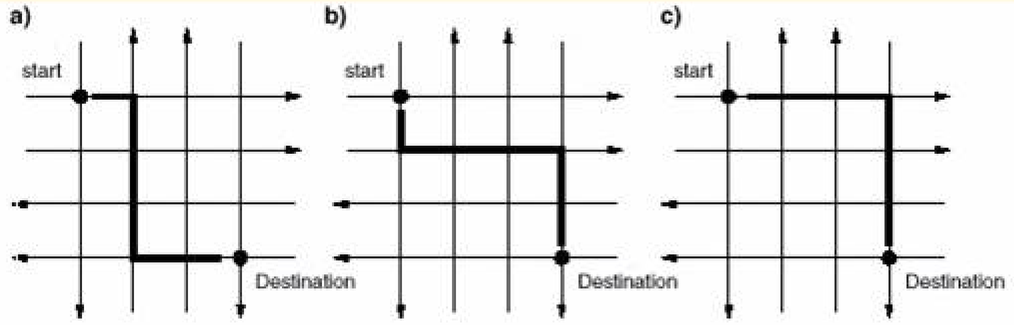

Fig 1. a) An illegal path. b) A legal but non-simple path. c) A simple path.

Input

On the first line of the input, there is a single integer n, telling how many city descriptions that follows. Each city description begins with a line containing three integers: the number of streets $0 < S \leq 30$ and avenues $0 < A \leq 30$ in the street grid, and the number of routes $0 < m \leq 200$ that should have at least one simple path. The next m lines define these routes, one on each line. Each route definition consists of four integers, $s1$, $a1$, $s2$, $a2$, where the start location of the route is at the crossing of street $s1$ and avenue $a1$, and the goal location is at the crossing of street $s2$ and avenue $a2$. Obviously, $0 < s1, s2 \leq S$ and $0 < a1, a2 \leq A$.

Output

For each city, your program should output 'Yes' on a single line if it is possible to make the streets and avenues one-way, so that each route has at least one simple path. Otherwise the text 'No' should be printed on a line of its own.

Sample Input

```
3
6 6 2
1 1 6 6
6 6 1 1
7 7 4
1 1 1 6
6 1 6 6
6 6 1 1
4 3 5 1
9 8 6
2 2 4 4
4 5 3 2
3 4 2 2
3 2 4 4
4 5 2 2
2 1 3 4
```

Sample Output

```
Yes
No
No
```