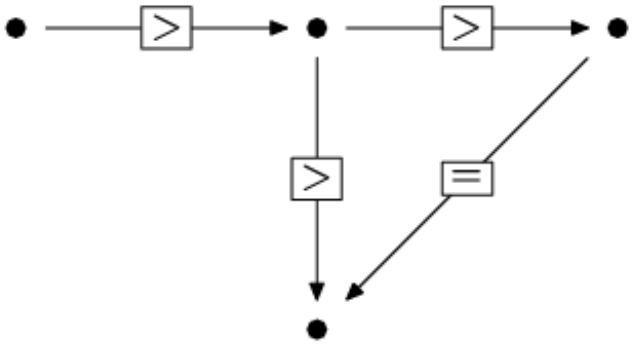
Relations

You are given a directed graph, whose edges are labeled with relational symbols '<', '>' and '='. For a nonnegative integer k, a k-correct G-labeling is a mapping from vertices of G into integers from interval [0,k] such that numbers at the ends of each edge satisfy the relation described by the label of the edge. We assume that an element on the left side of the relational symbol is a number assigned to the initial vertex. Compute the smallest k for which k-correct G-labeling exists or verify that such labeling doesn't exist for any k.

Illustration

For the graph in the figure the smallest k = 2.



Task

Write a program that for each data set from a sequence of several data sets:

- reads a description of a graph G from the input file,
- verifies whether there exist an integer k for which it is possible to label G k-correctly and, if the answer is positive, computes the smallest such k,
- writes the result to the output file.

Input

The first line of the input file contains one positive integer d not larger than 10. This is the number of data sets. The data sets follow. Each data set is described in two consecutive lines of the input file. In the first line there are two integers n and m separated by a single space. The number n is the number of vertices of G and m is the number of edges of G. Numbers n and m satisfy the inequalities: 1 <= n <= 1000, 0 <= m <= 10000. The vertices are numbered with integers from 1 to n and are identified by these numbers. There are no parallel edges and self-loops in the graph. (Two different edges u_1 -> v_1 and u_2 -> v_2 are parallel iff $u_1 = u_2$ and $v_1 = v_2$.) There are 3m integers separated by single spaces in the second line. The numbers at positions 3i-2 and 3i-1, 1 <= i <= m, are the ends of the i-th edge, the beginning and the end, respectively, whereas the number at position 3i is a number from the set {-1,0,1} and it is the label of the i-th edge: -1 represents '<', 0 represents '=' and 1 represents '>'.

Output

For the i-th data set, 1 <= i <= d, your program should write one word NO in the i-th line of the output file if a k-correct labeling doesn't exist for any k, or the smallest integer k for which such a labeling exists.

Example

Sample input:

```
4
4 4
1 2 -1 2 3 0 2 4 -1 3 4 -1
2 2
1 2 -1 2 1 -1
2 2
1 2 -1 2 1 1
3 3
1 2 0 3 2 0 3 1 0
```

Sample output:

```
2
NO
1
0
```