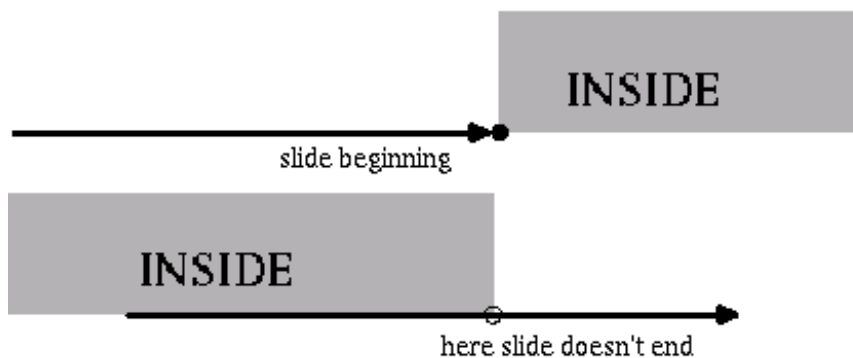


Icerink

A skating competition was organized on the largest icerink in Byteland. The icerink is a square of size $10000 * 10000$. A competitor begins skating at the START point chosen by referees and his task is to finish sliding at the FINISH point, also chosen by referees. The points of START and FINISH are different. One can slide in directions parallel to the sides of the icerink. There are some obstacles placed on the icerink. Each obstacle is a prism, which base is a polygon with sides parallel to the sides of the icerink. Each two adjacent sides of the base are always perpendicular. The obstacles do not have common points.

Each slide finishes up at the point where a competitor, for the first time, meets the wall of an obstacle, which is perpendicular to the direction of the slide. In other words, one can stop only when he crashes on a wall or in the FINISH point. Falling out of the icerink causes disqualification. Competitor may slide along walls of an obstacle.



Decide, whether a competitor who slides according to the given rules may reach the finish point, assuming he begun sliding from the starting point. If so, what is the minimal number of slides he needs to do?

Task

Write a program which:

- reads the description of the icerink, obstacles, and the coordinates of the start and finish point from the standard input,
- verifies, whether a competitor who begins from the starting point and slides according to the rules may reach the finish point, and if so, computes the minimal number of slides he needs to do,
- writes the result in the standard output.

Input

The number of test cases t is in the first line of input, then t test cases follow separated by an empty line. We define a system of coordinates to describe positions of objects on a rink. The rink is a square with vertices $(0, 0)$, $(10000, 0)$, $(10000, 10000)$, $(0, 10000)$. In the first line of each test case there are two integers z_1 and z_2 separated by a single space, $0 \leq z_1, z_2 \leq 10000$. The pair (z_1, z_2) denotes coordinates of the START point. In the second line of the file there are two integers t_1 and t_2 separated by single space, $0 \leq t_1, t_2 \leq 10000$. The pair (t_1, t_2) denotes

coordinates of the FINISH point. The third line of the file contains one integer s , $1 \leq s \leq 2500$. This is the number of obstacles. The following lines comprise descriptions of s obstacles. Each description of an obstacle begins with the line containing one positive integer r equal to the number of walls (sides of the base) of the obstacle. In each of the following r lines there are two integers x and y separated by a single space. These are the coordinates of the vertices of the obstacle's base, given in a clockwise order. (i.e. when going around the obstacle in this direction the inside is on the left-hand side). The total number of side walls of the obstacles does not exceed 10000.

Output

Your program should write for each test case:

- either one word 'NO' if it's impossible to get from the START point to the FINISH point
- or the minimal number of slides necessary to get to the FINISH point, if it is possible.

Example

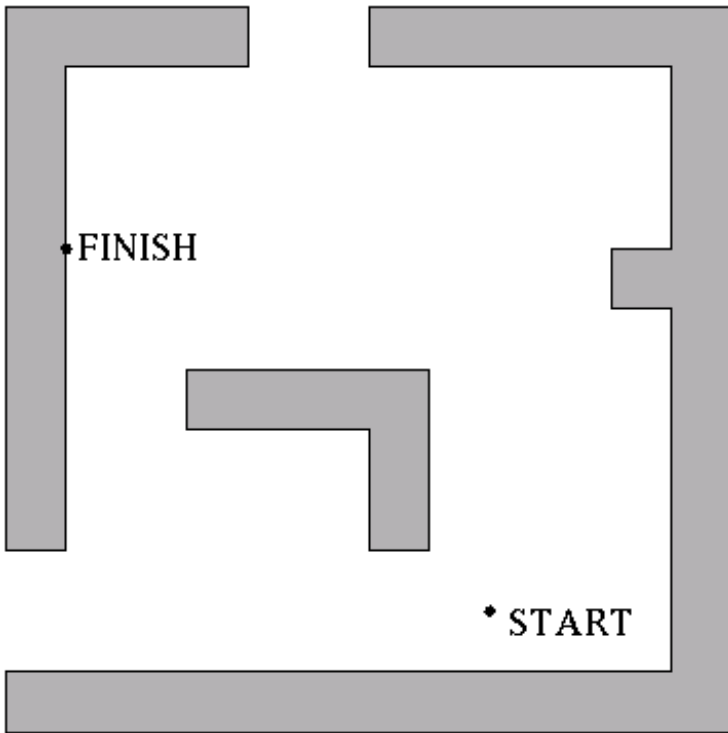
Sample input:

```
1
40 10
5 40
3
6
0 15
0 60
20 60
20 55
5 55
5 15
12
30 55
30 60
60 60
60 0
0 0
0 5
55 5
55 35
50 35
50 40
55 40
55 55
6
30 25
15 25
15 30
35 30
35 15
30 15
```

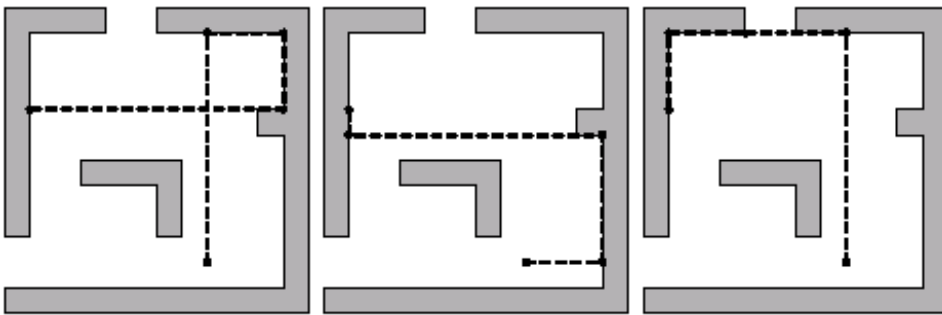
Sample output:

```
4
```

The sample input corresponds to the following situation:



These are the possible sequences of slides of length 4:



Warning: large Input/Output data, be careful with certain languages