



Task 4: OBELISK

A construction site consists of K floors, each of which can be treated as an infinitely large grid. On each floor, except the bottom floor, there are a number of holes, each one square large. At the bottom floor, there is a grid that is marked X. There is a heavy rectangular obelisk, of size $1 \times 1 \times M$, initially placed upright on the topmost floor. The construction workers would like to erect the obelisk vertically, such that one of the 1×1 faces is placed on the marked grid.

The obelisk is too heavy to be pushed or lifted, so the only way for the workers to move the obelisk is to roll it by tilting it along one of the edges as shown in Diagram 1.

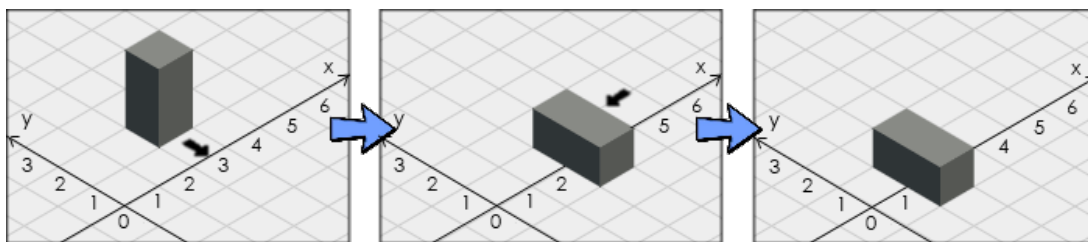


Diagram 1: Rolling an obelisk of size $1 \times 1 \times 2$

To move the obelisk from one floor to another floor below it, the workers will have to position the obelisk onto a hole, such that it falls to a lower floor, as shown in Diagram 2.

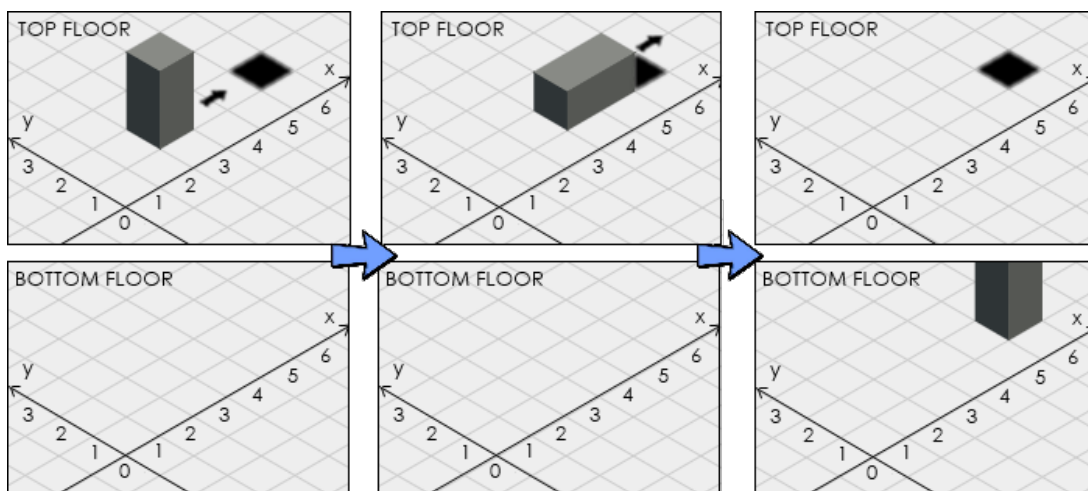


Diagram 2: Obelisk falling through a hole

As previously mentioned, each hole is 1×1 in size. No two holes are adjacent on the same floor. Hence, if the obelisk is horizontally placed over a hollow grid, it will not fall, unless M is 1. However, two or more holes in different layers may be vertically aligned – in this case the obelisk will fall through more than one layer, until it lands on a non-hole grid.

The obelisk is initially placed such that its edges are aligned with the grid in the construction area. Initially, the obelisk will always be placed vertically upright, with one of its 1×1 faces on a grid that is not a hole.

Your task is to find the the least number of moves to move the obelisk from its initial position on the top floor to the grid marked X on the bottom floor.



Input format

Your program must read from the standard input. The first line of the input contains 2 integers K and M , representing the number of floors and the height of the obelisk respectively.

The second line of the input contains four integers S_x, S_y, E_x, E_y , representing the starting x and y coordinates of the obelisk on the top floor and the ending x and y coordinates of the grid marked X on the bottom floor respectively.

Each of the next $K - 1$ lines describes the holes on each floor, starting from the top floor, down to the 2nd floor (i.e. the floor above the bottom floor). Each line begins with a positive integer h , indicating the number of holes on that floor, followed by a sequence of $2h$ space-separated integers $x_1, y_1, x_2, y_2, \dots, x_h, y_h$. For each i , the integers x_i and y_i indicate the x and y coordinates of a hole on that floor. Note that every floor except the bottom floor has at least one hole.

An example is provided below:

```
2 2
3 4 3 2
3 3 3 1 2 5 1
```

Output format

Your program must write to the standard output a single integer, which is either the minimum number of moves needed to erect the given obelisk on the grid marked as X, or -1 if it is not possible to move the obelisk onto the grid. Note that this answer may not fit in a 32-bit integer, requiring the use of long long in C/C++ or LongInt in Pascal. For the above example, the output is:

```
6
```

Subtasks

The maximum execution time on an input instance is 1.0s. Your program will be tested on 4 sets of input instances as follow:

1. (5 points) All instances in this set satisfy $M = 1, 1 \leq K \leq 10$. All coordinates are positive integers no more than 50. There are at most 10 holes on each floor.
2. (7 points) All instances in this set satisfy $1 \leq M \leq 50, 1 \leq K \leq 10$. All coordinates are positive integers no more than 50. There are at most 100 holes on each floor.
3. (6 points) All instances in this set satisfy $1 \leq M \leq 400, 1 \leq K \leq 500$. All coordinates are positive integers no more than 400. There are at most 100 holes on each floor.
4. (7 points) All instances in this set satisfy $1 \leq M \leq 100000, 1 \leq K \leq 500$. All coordinates are positive integers no more than 1,000,000,000. There are at most 100 holes on each floor.