

# Problem H

## Security Badge


**Problem ID:** securitybac

**CPU Time limit:** 3 seconds

**Memory limit:** 1024 MB

**Author:** David van Brack

**Source:** 2017 ICPC Mid-Atlantic Regional Contest

**License:** 

The home office of Labyrinthian Inc. has installed a new system of security badges and electronic door locks. Each badge is assigned an ID number, and the idea was that electronic locks on each door should allow access only to personnel whose ID number indicated that they had appropriate security clearance to enter that room, hallway, closet, or whatever lay on the other side of the door.



The contract for the lock system, however, was put out to the lowest bidder, who clearly misunderstood the intention. Instead of each lock storing a list of permitted ID numbers, instead each lock stores exactly two numbers, a lower and upper bound, and permits passage to badges whose number lies between those bounds. For example, a lock keyed as (25, 29) would pass only badges 25, 26, 27, 28, and 29.

Complicating the matter is the fact that lock on each side of the door can be keyed differently, so a person who is able to pass through the door in one direction might not be able to return once the door has closed behind them.

The results have been frustrating (and occasionally entertaining – videos of everyone in the company trying to find a way to the cafeteria at noon have gone viral on social media).

It has become a major task, when hiring or promoting any employee, to find a badge number that will actually get them from the front door to their new office.

Write a program to determine how many badge numbers will permit passage from one given room to another.

### Input

The first line of input will contain integers  $N$ ,  $L$ , and  $B$ , denoting the number of rooms, of locks, and of badge numbers, respectively.  $2 \leq N \leq 1\,000$ ,  $1 \leq L \leq 5\,000$ ,  $1 \leq B \leq 10^9$

The next line of input will contain two integers,  $S$  and  $D$ ,  $1 \leq S \leq N$ ,  $1 \leq D \leq N$ ,  $S \neq D$ , denoting the starting and destination rooms that we are interested in.

This is followed by  $L$  lines, each describing a lock as four integers:

$$a \ b \ x \ y$$

indicating that a lock permits passage from room  $a$  to room  $b$  (but not from  $b$  to  $a$ ) for badges numbered from  $x$  to  $y$ , inclusive. It is guaranteed that  $1 \leq a, b \leq N$ ,  $a \neq b$ ,  $1 \leq x \leq B$ ,  $1 \leq y \leq B$ ,  $x \leq y$ , and no  $(a, b)$  pair will occur more than once, although both  $(a, b)$  and  $(b, a)$  may occur within separate lines.

### Output

Print a single line indicating the number of badge ID numbers permitting passage from room  $S$  to room  $D$

#### Sample Input 1

```
4 5 10
3 2
1 2 4 7
3 1 1 6
3 4 7 10
2 4 3 5
4 2 8 9
```

#### Sample Output 1

```
5
```

#### Sample Input 2

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

#### Sample Output 2

```
5
```

