

Clear the Memory

You just bought a new computer with a memory chip that consists of M bits. The bits are numbered from 1 to M , inclusively, and can take one of two values, 0 or 1. All the bits of the memory chip are initially set to the value 1.

You turn on your computer and you want to allocate a large chunk of memory for your first program. A memory chunk can be allocated only if:

- It starts from position 1.
- It consists of consecutive bits.
- It consists entirely of bits with the value 0.

Unfortunately for you, you do not have direct access to change the values of the bits. Instead, you are given a list of N programs, numbered 1 to N , that you can run to affect the memory indirectly. Each of these programs comes with a user manual that describes which of the bits the program sets to 0 and which to 1 (the rest of the bits remain unaltered). You plan to execute a subset of the programs in some order of your choice to clear the largest possible chunk of memory that can be allocated.

You are asked to write a program to find what is the size of the largest chunk of memory that can be allocated.

Input

Your program must read from the standard input.

The first line of the input will contain three space-separated integer numbers N , M , and T : the number of programs available, the number of bits in the memory chip and the number of lines that contain the descriptions of the programs.

Each of the following T lines will contain four space-separated integer numbers P_i , A_i , B_i , and V_i , denoting that program P_i sets all the bits in the range $[A_i, B_i]$ to the value V_i .

A program may change multiple ranges of the memory to different values. It is guaranteed that for the same program these ranges will not overlap.

Output

Your program must print a single line to the standard output, consisting of a single

integer number: the length of the largest chunk of memory that can be allocated.

Constraints

- $1 \leq N \leq 5 \cdot 10^5$
- $1 \leq M \leq 5 \cdot 10^5$
- $1 \leq T \leq 5 \cdot 10^5$
- $1 \leq A_i \leq B_i \leq M$ and $0 \leq V_i \leq 1$ for all i
- Time and memory limit: See the CMS.

Subtasks

- Subtask 1 (7 points): $N \leq 6, M \leq 10^3$
- Subtask 2 (11 points): $M \leq 8$
- Subtask 3 (23 points): $N \leq 10^3, M \leq 10^3$
- Subtask 4 (28 points): $A_i = B_i$ for all i
- Subtask 5 (31 points): No further constraints.

Example

Input

```
3 6 5
1 2 4 0
2 1 3 0
2 5 5 1
3 5 5 0
3 2 2 1
```

Output

```
5
```

Explanation

All memory locations from bit 1 to bit 5, inclusively, can be set to 0 by first running program 2, then program 3 and finally program 1.

Actions	Memory
initially	111111
after 2 runs	000111
after 3 runs	010101
after 1 runs	000001

No program clears bit 6, so it is not possible to allocate a memory chunk of size larger than 5.