And the Rain Continues to Come Down (rain)

You are standing on a street of length N, initially positioned at K (positions are numbered from 0 to $N-1$). Throughout the day, rain will fall in certain areas of the street. It will rain M times, and for each rainfall, you are given four values:

- S_i: The time when the rain starts.
- E_i: The time when the rain ends (it will still be raining at time E_i).
- L_i, R_i: The segment of the street where it will rain (from position L_i to position R_i **inclusive**).

For example, if $N = 5$, one rainfall might be defined as: $S = 5$, $E = 12$, $L = 2$, $R = 4$. This means that from time 5 to 12 (inclusive), positions 2, 3, and 4 will be affected by rain.

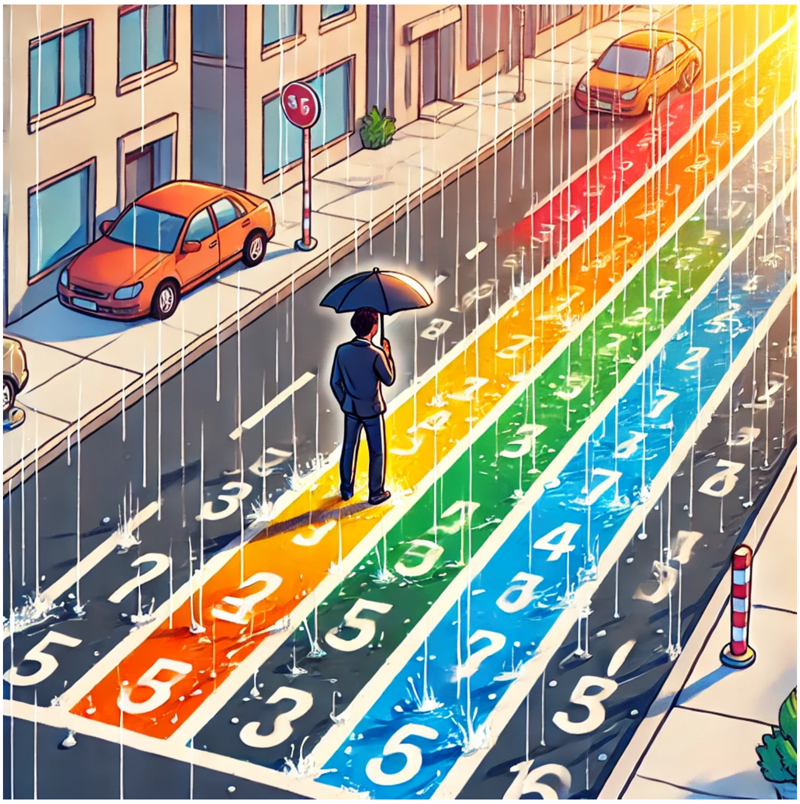

Figure 1: Sometimes it is difficult to stay dry.

Since you don't want to get wet, you must avoid the rain by teleporting to different positions on the street.

- At each unit of time (strictly after it starts and before it ends), you can teleport from your current position X to any other position Y at a cost of $|X - Y|$.
- However, you can only teleport if no position between X and Y is currently experiencing rain.

Calculate the **minimum total cost** required to avoid getting wet for the entire day. If it is impossible to avoid the rain, output that it cannot be done.

☞ Among the attachments of this task you may find a template file `rain.*` with a sample incomplete implementation.

Input

The first line of the input consists of three integers N, K, and M – the length of the street, your initial position, and the number of times it will rain.

Each of the following M lines of the input consists of four integers L_i, R_i, S_i, and E_i – it will rain at positions $L_i, L_i + 1, \ldots R_i$ during the time units $S_i, S_i + 1, \ldots, E_i$.

Output

Output a single line containing an integer, the minimum total cost to finish the day without getting wet, or -1 if it is impossible to do so.

Constraints

- $1 \leq N, M \leq 200\,000$.
- $0 \leq K < N$.
- $0 \leq L_i \leq R_i < N$ for each $i = 0 \ldots M - 1$.
- $0 \leq S_i \leq E_i \leq 10^9$ for each $i = 0 \ldots M - 1$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (15 points) $N \leq 1000$ and $M \leq 1000$.

– **Subtask 3** (10 points) $N \leq 1000$ and $E_i \leq 1000$.

– **Subtask 4** (45 points) There is no position affected by two different rainfalls in the same time unit. That is, there are no indices $0 \leq i < j < M$ such that the pairs of segments $[L_i, R_i]$ and $[L_j, R_j]$, and $[S_i, E_i]$ and $[S_j, E_j]$ both intersect.

– **Subtask 5** (30 points) No additional constraints.

Examples

input	output
5 1 3 1 2 1 5 0 3 3 3 4 4 5 5	4

input	output
5 1 3 1 2 1 5 3 4 3 3 3 4 4 5	1
5 1 3 1 2 1 5 0 3 3 3 4 4 4 5	-1
100 50 1 40 60 0 1	-1

Explanation

In the **first sample case**, an optimal sequence of moves is as follows:

- In time unit 0, move from position 1 (the initial position) to position 3 at the cost of 2.
- In time unit 2, move from position 3 to position 4 at the cost of 1.
- In time unit 4, move from position 4 to position 3 at the cost of 1.

At the end of the day, the total cost is $2 + 1 + 1 = 4$.

In the **second sample case**, it is enough to move from position 1 (the initial position) to position 0 in time unit 0. The cost of this teleport is 1, and it is possible to stay there until the end of the day.

In the **third sample case**, it is impossible to finish the day without being under the rain in at least one time unit. Notably, there is rainfall at position 3 that ends at time 3, and another rainfall at position 4 that begins at time 4. This means that attempting to teleport from position 4 to position 3 at either time 3 or time 4 while staying dry is impossible.

In the **fourth sample case**, the starting position is affected by rain in time unit 0, so it is impossible to avoid getting wet.