Stable Configuration

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

After founding the city and its rituals, the people turned their gaze to the deeper order hidden in the flow of light.

They carved shapes that followed its path and studied the patterns, seeking the rules of harmony woven within.

But they soon discovered that not all patterns brought harmony. Some fell out of balance and collapsed. To find balance again, they studied and refined their designs, over and over.

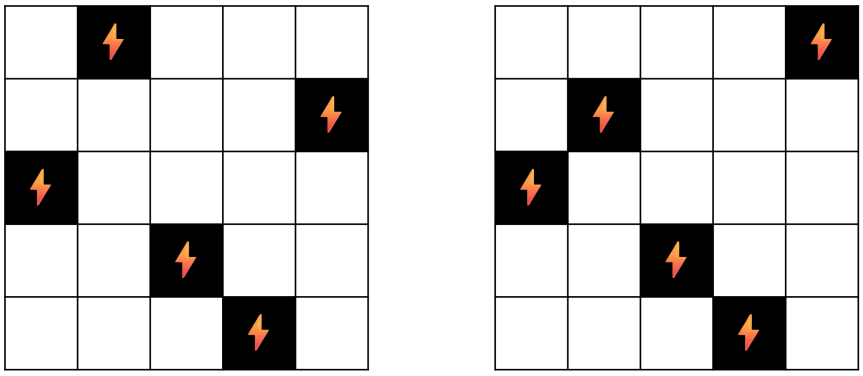
Retrace the arrangements they left behind, and restore the patterns of harmonious light.

The islanders conducted experiments by placing lights on an $N \times N$ grid, seeking to identify stable configurations. The cell at the r-th row from the top and c-th column from the left is denoted as (r, c).

A configuration is considered **stable** if it meets the following conditions:

- There is exactly one light in each row and exactly one light in each column.
- No unstable pattern exists.
  - An **unstable pattern** occurs when there are three lights located at positions $(r_1, c_1), (r_2, c_2), (r_3, c_3)$ such that $r_1 < r_2 < r_3$ and $c_1 > c_2 > c_3$.

For example, in the illustration below, the configuration on the left is stable. The one on the right is unstable because the three cells $(1, 5)$, $(2, 2)$, $(3, 1)$ form an unstable pattern.



To explore more refined arrangements, the islanders introduced additional constraints of the following form:

- Among the first c columns (from column 1 to column c), the maximum row index among the c placed lights must be exactly r.

They were interested in observing how the number of stable configurations would change as these constraints were added or removed.

Input

The first line contains two space-separated integers N and Q — the size of the grid and the number of queries, respectively.

Each of the following Q lines contains a query in one of the following formats:

- 1 r_i c_i: Add a new constraint. Among the c_i lights placed in columns 1 through c_i, the maximum row index must be exactly r_i.
- 2 x_i: Remove the constraint that was added as the x_i-th query.
- 3: Output the number of stable structures that satisfy all currently active constraints.

You may assume that there are no constraints before the queries.

Let t_i denote the type of the i-th query.

- $3 \leq N \leq 3 \times 10^5$
- $1 \leq Q \leq 3 \times 10^5$
- $1 \leq t_i \leq 3$ $(1 \leq i \leq Q)$
- $1 \leq r_i \leq N$ $(1 \leq i \leq Q, t_i = 1)$
- $1 \leq c_i \leq N$ $(1 \leq i \leq Q, t_i = 1)$
- $1 \leq x_i < i$ $(1 \leq i \leq Q, t_i = 2)$
- $t_{x_i} = 1$ $(1 \leq i \leq Q, t_i = 2)$
- $x_i \neq x_j$ $(1 \leq i < j \leq Q, t_i = t_j = 2)$
- There is at least one query of type 3.

Output

For each query of type 3, output a single line containing the number of stable structures that satisfy all currently active constraints, modulo $10^9 + 7$.

Scoring

- Subtask 1 (4 points): $N = 3, Q \leq 400$
- Subtask 2 (7 points): $N \leq 8, Q \leq 400$
- Subtask 3 (10 points): $Q = 1$
- Subtask 4 (23 points): $N \leq 400, Q \leq 400$
- Subtask 5 (25 points): $N \leq 2000, Q \leq 2000$
- Subtask 6 (13 points): $Q \leq 2$
- Subtask 7 (18 points): No additional constraints.

Examples

standard input	standard output
3 8 3 1 1 1 3 2 2 1 3 1 3 1 1 3 3	5 2 1 0
5 11 1 4 3 1 5 2 3 2 2 3 1 3 1 1 3 2 1 5 5 3 2 6 3	0 18 2 6
20 1 3	564120378
20 2 1 15 10 3	936990054