Algorithm Design Laboratory with Applications

Prof. Stefano Leucci

Problem: *Switches.*

Consider the following puzzle: nm lightbulbs are arranged in a $n \times m$ matrix. The generic lightbulb on row i and column j can be either WORKING or BROKEN. If it is BROKEN then it is also OFF, while if it is WORKING it can be either ON or OFF.

Next to each row $i = 1, \dots, n$ of the matrix there is a switch r_i. Similarly, each column $j = 1, \dots, m$ has a switch c_j. Flipping r_i (resp. c_j) will change the state of all WORKING lightbulbs on row i (resp. column j) from OFF to ON and vice-versa.

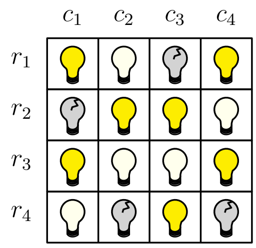
Design an algorithm that, given n, m, and the initial state of every lightbulb, decides whether there is a subset of switches to flip that will cause all the lightbulbs to be OFF.

Input. The input consists of a set of instances, or *test-cases*, of the previous problem. The first line contains the number T of test-cases. The first line of each test case contains three integers n, m, and t where t is the number if WORKING lightbulbs. Each of the following t lines corresponds to a WORKING lightbulb and contains the integers i, j, s corresponding to the lightbulb row, column and state, where $s = 0$ if the lightbulb is OFF and $s = 1$ if it is ON.

Output. The output consists of T lines. The i-th line is the answer to the i-th test-case and contains “YES” if it is possible to light up all lightbulbs simultaneously and “NO” otherwise.

Assumptions. $1 \leq T \leq 10$; $1 \leq n \leq 2^{11}$; $1 \leq m \leq 2^{11}$.

Example.



Input (corresponding to the above picture):

```
1
4 4 12
1 1 1
1 2 0
1 4 1
2 2 1
2 3 1
2 4 0
3 1 1
3 2 0
3 3 0
3 4 1
4 1 0
4 3 1
```

Output (corresponding to subset of switches $\{r_2, r_4, c_1, c_4\}$):

```
YES
```

Requirements. Your algorithm should require $O(n + m + t)$ time (with reasonable hidden constants).

Notes. A reasonable implementation should not require more than 3 seconds for each input file.