Inverted World

Problem B

Luo Tsz Fung {pepper1208}

The 1st Buddhist Sin Tak College Computer Club Programming Contest

October 18, 2024

stpc();

Background

Problem Idea by `pepper1208`
Preparation by `pepper1208`, `rina__owo`

Problem Restatement

Given a $N \times N$ 0-grid initially.

After Q inversion of several rows and several columns, count the number of 1s inside the grid.

Inversion: Change 0 to 1, or change 1 to 0.

Statistics

Points are given per subtask in this problem. You have to pass all the checkpoint in the subtask in order to get the points of the subtask.

Attempts: 27

0 points	12	+	0	=	12
Subtask 1 (5 points)	0	+	2	=	2
Subtask 2 (6 points)	0	+	2	=	2
Subtask 3 (26 points)	0	+	2	=	2
Subtask 4 (38 points)	0	+	2	=	2
Subtask 5 (25 points)	0	+	0	=	0

First solved by **No one!**

Subtasks

Subtask	Score	N	Q
1	5	$= 2$	≤ 1
2	6	$= 3$	≤ 10
3	26	≤ 1000	≤ 10
4	38	≤ 1000	$\leq 10^6$
5	25	$\leq 10^6$	$\leq 10^6$

Subtask 1 (5 pts): $N = 2, Q \leq 1$

Sanity check.

If $Q = 0$, output 0.
If $Q = 1$, output 2.

Score: **5** (Cumulative Score: **5**)

Subtask 2 - 4 (70 pts): $N \leq 1000, Q \leq 10^6$

These subtasks could be passed by pure simulation.

First, you can build a $N \times N$ integer 2-dimensional array. The array will be initialised by 0.

Then, for each action, invert the whole row / column by a for loop.

Subtask 2 - 4 (70 pts): $N \leq 1000, Q \leq 10^6$

You can simply invert a number in `arr[i][j]` by `arr[i][j] = 1 - arr[i][j]`.

Invert the whole row = Invert from `arr[i][1]` to `arr[i][N]` (for 1-based array.)
The logic can be similarly applied to invert the whole column.

Score: **70** (Cumulative Score: **75**)

Full Solution

To achieve the full solution, you need two main observations for this problem.

Observation 1

All inversions are **independent** from each other.

The observation deduces that all actions will not affect the others. For example, the order of the actions will not affect the final result.

Full Solution

Observation 2

If there are k actions which invert the same row/column, the required row/column are only inverted by k **mod 2** times only.

As all inversion as independent from each other, then inverting the same row/column **twice** is equal to remain the required row/column **unchanged**.

Full Solution

Therefore, we can implement **two arrays with size** N (or $N \times 2$ array as you like) to store the **frequency of inversion** of each row/column.

Denote r and c be the number of valid inversion to rows and columns respectively.

Find r by linear searching an array storing the frequency of inversion of rows. The answer now should be Nr.

Then, find c using the similar method as above. The answer now should be

$$Nr + Nc - 2rc$$

Score: **25** (Cumulative Score: **100**)

Takeaways

1. For normal contestants: grab as many points as possible using simple method(s)!
2. For advanced contestants: observation, observation and observation is the key to success.