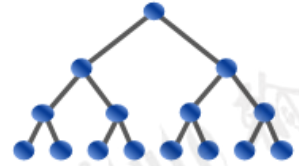


# USA Computing Olympiad



OVERVIEW

DETAILS / FAQ

TRAINING

HISTORY

RESOURCES

## USACO 2024 DECEMBER CONTEST, SILVER PROBLEM 3. 2D CONVEYOR BELT

[Return to Problem List](#)

Contest has ended.

[Log in to allow submissions in analysis mode](#)

English (en) ▼

Farmer John's milk factory can be described by an  $N$  by  $N$  ( $1 \leq N \leq 1000$ ) grid of cells that contain conveyor belts. Position  $(a, b)$  describes the cell that is in the  $a$ -th row from the top and  $b$ -th column from the left. There are 5 types of cells:

1. "L" — the cell is a leftward facing conveyor belt which moves all items on it 1 cell left every time unit.
2. "R" — the cell is a rightward facing conveyor belt which moves all items on it 1 cell right every time unit.
3. "U" — the cell is an upward facing conveyor belt which moves all items on it 1 cell up every time unit.
4. "D" — the cell is a downward facing conveyor belt which moves all items on it 1 cell down every time unit.
5. "?" — Farmer John has not built a conveyor belt at that cell yet.

Note that conveyor belts can also move items outside the grid. A cell  $c$  is *unusable* if an item placed at cell  $c$  will **never** exit the conveyor belt grid (i.e. it will move around in the grid forever).

Initially, Farmer John has not started building the factory so all cells start out as "?". For the next  $Q$  ( $1 \leq Q \leq 2 \cdot 10^5$ ) days starting from day 1 and ending at day  $Q$ , Farmer John will choose a cell that does **not** have a conveyor belt and build a conveyor belt at the cell.

Specifically, during the  $i$ -th day, Farmer John will build a conveyor belt of type  $t_i$  ( $t_i \in \{L,R,U,D\}$ ) at position  $(r_i, c_i)$  ( $1 \leq r_i, c_i \leq N$ ). It is guaranteed that there is no conveyor belt at position  $(r_i, c_i)$ .

After each day, help Farmer John find the minimum number of unusable cells he can achieve by **optimally** building conveyor belts on all **remaining** cells without a conveyor belt.

### INPUT FORMAT (input arrives from the terminal / stdin):

The first line contains  $N$  and  $Q$ .

The  $i$ -th of the next  $Q$  lines contains  $r_i, c_i$ , and  $t_i$  in that order.

### OUTPUT FORMAT (print output to the terminal / stdout):

$Q$  lines, the  $i$ -th of which describing the minimum number of unusable cells if Farmer John fills optimally builds conveyor belts on all remaining cells that do not currently have a conveyor belt.

### SAMPLE INPUT:

```

3 5
1 1 R
3 3 L
3 2 D
1 2 L
2 1 U

```

### SAMPLE OUTPUT:

```

0
0
0
2
3

```

The conveyor belt after the fifth day is shown below.

```

RL?
U??
?DL

```

One optimal way to build conveyor belts on the remaining cells is as follows.

```

RLR
URR
LDL

```

In this configuration, the cells at  $(1, 1)$ ,  $(1, 2)$ , and  $(2, 1)$  are unusable.

**SAMPLE INPUT:**

```

3 8
1 1 R
1 2 L
1 3 D
2 3 U
3 3 L
3 2 R
3 1 U
2 1 D

```

**SAMPLE OUTPUT:**

```

0
2
2
4
4
6
6
9

```

The conveyor belt after the eighth day is shown below.

```

RLD
D?U
URL

```

No matter what conveyor belt Farmer John can build at the center, all cells will be unusable.

**SAMPLE INPUT:**

```

4 13
2 2 R
2 3 R
2 4 D
3 4 D
4 4 L
4 3 L
4 2 U
3 1 D
4 1 R
2 1 L
1 1 D
1 4 L
1 3 D

```

**SAMPLE OUTPUT:**

```

0
0
0
0
0
0
0
0
0
11
11
11
11
13

```

**SCORING:**

- Inputs 4-5:  $N \leq 10$
- Inputs 6-7:  $N \leq 40$
- Inputs 8-13: No additional constraints

Problem credits: Alex Liang

Contest has ended. No further submissions allowed.

