



International Olympiad in Informatics

2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

game

English — 1.1

Bazza and Shazza are playing a game. The board is a grid of cells, with R rows numbered $0, \dots, R - 1$, and C columns numbered $0, \dots, C - 1$. We let (P, Q) denote the cell in row P and column Q . Each cell contains a non-negative integer, and at the beginning of the game all of these integers are zero.

The game proceeds as follows. At any time, Bazza may either:

- update a cell (P, Q) , by assigning the integer that it contains;
- ask Shazza to calculate the greatest common divisor (GCD) of all integers within a rectangular block of cells, with opposite corners (P, Q) and (U, V) inclusive.

Bazza will take $N_U + N_Q$ actions (updating cells N_U times and asking questions N_Q times) before he gets bored and goes outside to play cricket.

Your task is to work out the correct answers.

Example

Suppose $R = 2$ and $C = 3$, and Bazza begins with the following updates:

- Update cell $(0, 0)$ to 20;
- Update cell $(0, 2)$ to 15;
- Update cell $(1, 1)$ to 12.

20	0	15
0	12	0

The resulting grid is shown in the picture above. Bazza might then ask for GCDs in the following rectangles:

- Opposite corners $(0, 0)$ and $(0, 2)$: The three integers in this rectangle are 20, 0 and 15, and their GCD is 5.
- Opposite corners $(0, 0)$ and $(1, 1)$: The four integers in this rectangle are 20, 0, 0 and 12, and their GCD is 4.

Now suppose Bazza makes the following updates:

- Update cell $(0, 1)$ to 6;
- Update cell $(1, 1)$ to 14.

20	6	15
0	14	0

The new grid is shown in the picture above. Bazza might then ask for GCDs in the following rectangles again:

- Opposite corners $(0, 0)$ and $(0, 2)$: Now the three integers in this rectangle are 20, 6 and 15, and their GCD is 1.
- Opposite corners $(0, 0)$ and $(1, 1)$: Now the four integers in this rectangle are 20, 6, 0 and 14, and their GCD is 2.

Here Bazza has performed $N_U = 5$ updates and $N_Q = 4$ questions.

Implementation

You should submit a file implementing the procedures `init()` and `update()` and the function `calculate()`, as described below.

To help you, the template solutions provided on your computer (`game.c`, `game.cpp` and `game.pas`) each include a function `gcd2(X, Y)` to compute the greatest common divisor of two given non-negative integers X and Y . If $X = Y = 0$ then `gcd2(X, Y)` will return 0 also.

This function is fast enough to score full points; in particular, the running time is at worst proportional to $\log(X + Y)$.

Your Procedure: `init()`

C/C++ `void init(int R, int C);`

Pascal `procedure init(R, C : LongInt);`

Description

Your submission must implement this procedure.

This procedure gives you the initial size of the grid, and allows you to initialise any global variables and data structures. It will be called only once, before any calls to `update()` or `calculate()`.

Parameters

- `R` : The number of rows.
- `C` : The number of columns.

Your Procedure: `update()`

C/C++ `void update(int P, int Q, long long K);`

Pascal `procedure update(P, Q : LongInt; K : Int64);`

Description

Your submission must implement this procedure.

This procedure will be called when Bazza assigns the number in some grid cell.

Parameters

- `P` : The row of the grid cell ($0 \leq P \leq R - 1$).
- `Q` : The column of the grid cell ($0 \leq Q \leq C - 1$).
- `K` : The new integer in this grid cell ($0 \leq K \leq 10^{18}$). May be the same as the current value.

Your Function: calculate ()

C/C++ `long long calculate(int P, int Q, int U, int V);`

Pascal `function calculate(P, Q, U, V : LongInt) : Int64;`

Description

Your submission must implement this function.

This function should calculate the greatest common divisor of all integers in the rectangle with opposite corners (P, Q) and (U, V) . This range is inclusive, i.e., the cells (P, Q) and (U, V) are included in the rectangle.

If all integers in this rectangle are zero, then this function should return zero also.

Parameters

- P : The row of the top-left cell in the rectangle ($0 \leq P \leq R - 1$).
- Q : The column of the top-left cell in the rectangle ($0 \leq Q \leq C - 1$).
- U : The row of the bottom-right cell in the rectangle ($P \leq U \leq R - 1$).
- V : The column of the bottom-right cell in the rectangle ($Q \leq V \leq C - 1$).
- *Returns*: The GCD of all integers in the rectangle, or 0 if all of those integers are zero.

Sample Session

The following session describes the example above:

Function Call	Returns
<code>init(2, 3)</code>	
<code>update(0, 0, 20)</code>	
<code>update(0, 2, 15)</code>	
<code>update(1, 1, 12)</code>	
<code>calculate(0, 0, 0, 2)</code>	5
<code>calculate(0, 0, 1, 1)</code>	4
<code>update(0, 1, 6)</code>	
<code>update(1, 1, 14)</code>	
<code>calculate(0, 0, 0, 2)</code>	1
<code>calculate(0, 0, 1, 1)</code>	2

Constraints

- Time limit: See subtasks.
- Memory limit: See subtasks.
- $1 \leq R, C \leq 10^9$
- $0 \leq K \leq 10^{18}$, where `K` is any integer that Bazza places in a grid cell.

Subtasks

Subtask	Points	R	C	N_u	N_q	Time limit	Memory limit
1	10	≤ 100	≤ 100	≤ 100	≤ 100	13 s	230 MiB
2	27	≤ 10	$\leq 100,000$	$\leq 10,000$	$\leq 250,000$	13 s	230 MiB
3	26	$\leq 2,000$	$\leq 2,000$	$\leq 10,000$	$\leq 250,000$	13 s	230 MiB
4	17	$\leq 10^9$	$\leq 10^9$	$\leq 10,000$	$\leq 250,000$	13 s	230 MiB
5	20	$\leq 10^9$	$\leq 10^9$	$\leq 22,000$	$\leq 250,000$	13 s	230 MiB

Experimentation

The sample grader on your computer will read input from the file `game.in`. This file must be in the following format:

- line 1: `R C N`
- next `N` lines: one action per line, in the order in which actions occur

The line for each action must be in one of the following formats:

- to indicate `update(P, Q, K) : 1 P Q K`
- to indicate `calculate(P, Q, U, V) : 2 P Q U V`

For instance, the example above should be provided in the following format:

```
2 3 9
1 0 0 20
1 0 2 15
1 1 1 12
2 0 0 0 2
2 0 0 1 1
1 0 1 6
1 1 1 14
2 0 0 0 2
2 0 0 1 1
```

Language Notes

C/C++ You must `#include "game.h"`.

Pascal You must define the `unit Game`. All arrays are numbered beginning at `0` (not `1`).

Because the integers in grid cells could be very large, C/C++ users are advised to use the `long long` type, and Pascal users are advised to use the `Int64` type.