



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

game

Czech — 1.1

Bazza a Shazza hrají hru *Naunuděnou*. Tato hra se hraje na mřížce s R řádky očíslovanými $0, \dots, R-1$ a C sloupci očíslovanými $0, \dots, C-1$. Na jednotlivých políčkách mřížky se nacházejí nezáporná celá čísla, která jsou na začátku všechna rovna nule. Pro zjednodušení zápisu budeme políčko v řádku P a sloupci Q označovat jako (P, Q) .

V každém tahu může Bazza udělat jednu z následujících dvou věcí:

- změnit číslo na políčku (P, Q)
- zeptat se Shazzy na nejvyšší společný dělitel (NSD) všech čísel v obdélníkové oblasti, jejíž protější rohy tvoří políčka (P, Q) a (U, V) (oblast obsahuje i tato dvě políčka).

Bazza odehraje $N_U + N_Q$ tahů, z nichž N_U jsou změny čísel na některých z políček a N_Q jsou dotazy na NSD, neboť pak se začne nudit a raději půjde hrát ven kriket. To, že Shazza se nudí celou dobu, asi není třeba explicitně zmiňovat.

Vaším úkolem je pomoci Shazze odpovídat na Bazzovy dotazy.

Příklad

Předpokládejte, že $R = 2$, $C = 3$ a že Bazza začal s následujícími tahy:

- Změň hodnotu políčka $(0, 0)$ na 20;
- Změň hodnotu políčka $(0, 2)$ na 15;
- Změň hodnotu políčka $(1, 1)$ na 12.

20	0	15
0	12	0

Po těchto tazích bude mřížka vypadat tak, jak je znázorněno na předchozím obrázku. Bazza pokračuje dvěma dotazy na NSD čísel v oblastech s následujícími protějšími rohy:

- $(0, 0)$ a $(0, 2)$: Tři čísla v této oblasti jsou 20, 0 a 15. Jejich NSD je 5.
- $(0, 0)$ a $(1, 1)$: Čtyři čísla v této oblasti jsou 20, 0, 0 a 12. Jejich NSD je 4.

Bazza pokračuje následujícími dvěma tahy:

- Změň hodnotu políčka $(0, 1)$ na 6;
- Změň hodnotu políčka $(1, 1)$ na 14.

20	6	15
0	14	0

Po těchto tazích bude mřížka vypadat tak, jak je znázorněno na předchozím obrázku. Bazza pokračuje dvěma dotazy na NSD čísel v oblastech s následujícími protějšími rohy:

- $(0, 0)$ a $(0, 2)$: Tři čísla v této oblasti jsou nyní 20, 6 a 15. Jejich NSD je 1.
- $(0, 0)$ a $(1, 1)$: Čtyři čísla v této oblasti jsou nyní 20, 6, 0 a 14. Jejich NSD je 2.

V tomto příkladu Bazza udělal $N_U = 5$ změn a $N_Q = 4$ dotazů.

Implementace

Odevzdejte soubor implementující funkce `init()`, `update()` a `calculate()`.

Abychom vám pomohli, kostry řešení ve složce úlohy (soubory `game.c`, `game.cpp` a `game.pas`) obsahují funkci `gcd2(X, Y)`, která počítá NSD zadaných čísel X a Y . Pokud $X = Y = 0$, pak `gcd2(X, Y)` vrátí 0.

Tato funkce je dostatečně rychlá k tomu, abyste dosáhli plného počtu bodů. Její časová složitost v nejhorším případě je $O(\log(X+Y))$.

Vaše procedura: `init()`

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

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

Popis

Naimplementujte tuto proceduru.

Tato procedura dostane velikost mřížky a umožní vám inicializovat globální proměnné a datové struktury. Bude zavolána pouze jednou před funkcemi `update()` a `calculate()`.

Parametry

- `R` : Počet řádků.
- `C` : Počet sloupců.

Vaše procedura: `update()`

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

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

Popis

Naimplementujte tuto proceduru.

Tato procedura je zavolána, když Bazza mění číslo na nějakém políčku mřížky.

Parametry

- `P` : Číslo řádku, ve kterém se políčko nachází ($0 \leq P \leq R - 1$).
- `Q` : Číslo sloupce, ve kterém se políčko nachází ($0 \leq Q \leq C - 1$).
- `K` : Nová hodnota čísla na změněném políčku ($0 \leq K \leq 10^{18}$). Může se stát, že nová hodnota bude stejná, jako předchozí hodnota na tomto políčku.

Vaše funkce: `calculate()`

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

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

Popis

Naimplementujte tuto funkci.

Tato funkce má spočítat NSD všech čísel na obdélníkové oblasti s protilehlými rohy na políčkách (P, Q) a (U, V) (políčka (P, Q) a (U, V) jsou součástí oblasti).

Pokud jsou všechna čísla v této oblasti nulová, funkce by měla vrátit nulu.

Parametry

- P : Číslo řádky levého horního rohu oblasti ($0 \leq P \leq R - 1$).
- Q : Číslo sloupce levého horního rohu oblasti ($0 \leq Q \leq C - 1$).
- U : Číslo řádku pravého dolního rohu oblasti ($P \leq U \leq R - 1$).
- V : Číslo sloupce pravého dolního rohu oblasti ($Q \leq V \leq C - 1$).
- *Návratová hodnota*: NSD všech čísel v zadané oblasti, nebo 0 když jsou všechna čísla v oblasti rovna nule.

Ukázka průběhu programu

Následující ukázka odpovídá uvedenému příkladu:

Volání funkce	Návratová hodnota
<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

Omezení

- Časový limit: viz podúlohy.

- Paměťový limit: viz podúlohy.
- $1 \leq R, C \leq 10^9$
- $0 \leq K \leq 10^{18}$ pro každé číslo K , které Bazza umístí do mřížky.

Podúlohy

Hodnoty parametrů pro podúlohy naleznete v anglické verzi.

Podúloha	Body	R	C	N_U	N_Q	Časový limit	Paměťový limit

Experimentování

Vzorový testovač čte soubor `game.in` v následujícím formátu:

- řádek 1: `R C N`
- následujících N řádků: jeden tah na každém z těchto řádků v pořadí, ve kterém je Bazza zahrál

Řádek každého tahu musí mít jeden z následujících formátů:

- změna políčka, tj. volání funkce `update(P, Q, K) : 1 P Q K`
- dotaz na NSD, tj. volání funkce `calculate(P, Q, U, V) : 2 P Q U V`

Uvedenému příkladu odpovídá následující obsah souboru:

```
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
```

Poznámky k programovacím jazykům

- C/C++ Na začátku souboru s řešením musí být řádek `#include "game.h"`.
- Pascal Naimplementujte `unit Game`. Položky všech polí jsou číslovány od `0` (tj. nikoliv od `1`).

Protože čísla v mřížce mohou být velmi velká, doporučujeme používat typ `long long` v jazyku C/C++ a `Int64` v Pascalu.