



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

game

Romanian — 1.1

Bazza și Shazza vor să joace un joc. Tabla este un grid de celule, cu R rânduri numerotate $0, \dots, R - 1$ și C coloane numerotate $0, \dots, C - 1$. Fie (p, q) celula din rândul p și coloana q . Fiecare celulă conține un întreg pozitiv, la începutul jocului toate acestea conținând zero.

Jocul se desfășoară în felul următor. În orice moment, Bazza poate:

- să facă un update la celula (p, q) , schimbând întregul pe care îl conține;
- să îi ceară lui Shazza să calculeze cel mai mare divizor comun (CMMDC) al tuturor întregilor dintr-o regiune dreptunghiulară de celule, având colțurile opuse (p, q) și (u, v) (inclusiv aceste celule).

Bazza va face cel mult $N_U + N_Q$ acțiuni (updateare celule de N_U ori și punere de întrebări de N_Q ori) până se plictisește și iese afară ca să se joace cricket.

Voi trebuie să calculați răspunsurile corecte.

Exemple

Presupunând că $R = 2$ and $C = 3$, și Bazza începe cu următoarele update-uri:

- Update la celula $(0, 0)$ la valoarea 20;
- Update la celula $(0, 2)$ la valoarea 15;
- Update la celula $(1, 1)$ la valoarea 12.

20	0	15
0	12	0

Grid-ul rezultat este afișat în imaginea de mai sus. Bazza poate să întrebe care este CMMDC-ul următoarelor zone dreptunghiulare:

- Colțuri opuse $(0, 0)$ and $(0, 2)$: cei trei întregi din acest dreptunghi sunt 20, 0 și 15, având CMMDC-ul 5.
- Colțuri opuse $(0, 0)$ and $(1, 1)$: cei patru întregi din acest dreptunghi sunt 20, 0, 0 și 12, având CMMDC-ul 4.

Presupunând că acum Bazza face următoarele update-uri:

- Update la celula $(0, 1)$ la valoarea 6;
- Update la celula $(1, 1)$ la valoarea 14.

20	6	15
0	14	0

Noul grid este afișat în imaginea de mai sus. Bazza poate întreba pentru CMMDC-urile următoarelor zone dreptunghiulare (din nou):

- Colțuri opuse $(0, 0)$ și $(0, 2)$: acum cei trei întregi în acest dreptunghi sunt 20, 6 și 15, având CMMDC-ul 1.
- Colțuri opuse $(0, 0)$ și $(1, 1)$: acum cei patru întregi din acest dreptunghi sunt 20, 6, 0 și 14, având CMMDC-ul 2.

Aici Bazza a performat $N_u = 5$ update-uri și $N_q = 4$ query-uri.

Implementare

Trebuie să submitați un fișier care implementează funcțiile `init()`, `update()` și `calculate()`, conform descrierii următoare.

Pentru a vă ajuta, soluțiile - template de pe calculatorul vostru (`game.c`, `game.cpp` și `game.pas`) conține fiecare o funcție `gcd2(X, Y)` pentru calculul celui mai mare divizor comun pentru două numere întregi nenegative X și Y . Dacă $X = Y = 0$ atunci `gcd2(X, Y)` va returna deasemenea `0`.

Această funcție este suficient de rapidă pentru a obține scorul maxim; mai precis, timpul de execuție este în cel mai rău caz proporțional cu `log(X + Y)`.

Procedura voastră: `init()`

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

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

Descriere

Submisia voastră trebuie să implementeze această funcție.

Această funcție vă furnizează dimensiunile inițiale ale gridului și vă permite să inițializați orice variabile sau structuri de date globale. Aceasta va fi apelată o singură dată înainte de orice apel pentru `update()` sau `calculate()`.

Parametri

- `R`: Numărul de linii.
- `C`: Numărul de coloane.

Procedura voastră: `update()`

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

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

Descriere

Submisia voastră trebuie să implementeze această funcție.

Această funcție va fi apelată atunci când Bazza schimbă valoarea într-o anumită celulă din grid.

Parametri

- `P`: Linia celulei de grid ($0 \leq P \leq R - 1$).
- `Q`: Coloana celulei de grid ($0 \leq Q \leq C - 1$).

- K : Valoarea nouă care va fi conținută de celula din grid ($0 \leq K \leq 10^{18}$). Poate fi la fel ca valoarea curentă.

Funcția voastră: `calculate()`

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

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

Descriere

Submisia voastră trebuie să implementeze această funcție.

Această funcție trebuie să calculeze cel mai mare divizor comun a tuturor întregilor din dreptunghiul cu colțurile în celulele de grid (P, Q) și (U, V) . Colțurile fac parte din dreptunghi.

Dacă toate valorile din dreptunghi sunt nule funcția trebuie să returneze zero.

Parametri

- P : Linia celulei stânga-sus a dreptunghiului ($0 \leq P \leq R - 1$).
- Q : Coloana celulei stânga-sus a dreptunghiului ($0 \leq Q \leq C - 1$).
- U : Linia celulei dreapta-jos a dreptunghiului ($P \leq U \leq R - 1$).
- V : Coloana celulei dreapta-jos a dreptunghiului ($Q \leq V \leq C - 1$).
- *Return*: Cel mai mare divizor comun al tuturor valorilor din dreptunghi sau 0 toate aceste valori sunt zero.

Exemplu

Aici aveți descris exemplul de mai sus:

Apelul Funcției	Returnează
<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

Constrângeri

- Limită de timp: vezi subtask-uri
- Limită de memorie: vezi subtask-uri
- $1 \leq R, C \leq 10^9$
- $0 \leq K \leq 10^{18}$, unde K este un întreg pe care Bazza îl plasează în grid.

Subtask-uri

Vezi versiunea în engleză pentru parametrii subtask-urilor.

Testare

Graderul de pe computerul vostru va citi input din fișierul `game.in`. Acest fișier trebuie să aibă următorul format:

- linia 1: `R C N`
- următoarele `N` linii: o acțiune pe linie, în ordinea în care apar

Linia pentru fiecare acțiune trebuie să fie una dintre următoarele formate:

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

Astfel, exemplul de mai sus ar fi dat în următorul 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
```

Note de limbaj

C/C++ Trebuie să faceți `#include "game.h"`.

Pascal Trebuie să definiți `unit Game`. Toți vectorii sunt indexați începând de la `0` (nu de la `1`).

Deoarece întregii din grid pot fi foarte mari, utilizatorii C/C++ sunt sfătuiți să folosească tipul de date `long long`, și utilizatorii Pascal sunt sfătuiți să folosească tipul de date `Int64`.