



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

game

Croatian — 1.0

Mirko i Slavko igraju igru. Ploča je sastavljena od R redaka i C stupaca. Retci i stupci su označeni cijelim uzastopnim brojevima od 0 .

Neka je (P, Q) polje u retku P i stupcu Q . Svako polje sadrži nenegativni cijeli broj. Na početku igre ploča je popunjena s nulama.

U svakom potezu Mirko može:

- promijeniti broj na polju (P, Q) ;
- pitati Slavka da mu izračuna najveću zajedničku mjeru (GCD) svih brojeva u pravokutnom bloku polja s nasuprotnim vrhovima u (P, Q) i (U, V) uključivo.

Mirko će napraviti $N_U + N_Q$ poteza (N_U promjena polja i N_Q upita) prije no što mu igra dosadi i ode van proučavati psiho-nautiku.

Vaš zadatak je pomoći Slavku da odigra igru.

Primjer

Neka je $R = 2$ i $C = 3$ i Mirko igra sljedeće promjene:

- Polje $(0, 0)$ na 20;
- Polje $(0, 2)$ na 15;
- Polje $(1, 1)$ na 12.

20	0	15
0	12	0

Rezultantna ploča je prikazana na slici iznad. Mirko sada može pitati GCD vrijednosti u sljedećim pravokutnicima:

- Nasuprotni vrhovi $(0, 0)$ i $(0, 2)$: Tri broja u ovom pravokutniku su 20, 0 i 15, stoga je rješenje 5.
- Nasuprotni vrhovi $(0, 0)$ i $(1, 1)$: Četiri broja u ovom pravokutniku su 20, 0, 0, i 12, stoga je rješenje 4.

Pretpostavimo da Mirko sada napravi sljedeće promjene:

- Polje $(0, 1)$ na 6;
- Polje $(1, 1)$ na 14.

20	6	15
0	14	0

Nova ploča je prikazana na slici iznad. Mirko sada može pitati sljedeće upite:

- Nasuprotni vrhovi $(0, 0)$ i $(0, 2)$: Tri broja u ovom pravokutniku su 20, 6 i 15, stoga je rješenje 1.
- Nasuprotni vrhovi $(0, 0)$ i $(1, 1)$: Četiri broja u ovom pravokutniku su 20, 6, 0, i 14, stoga je rješenje 2.

U ovom primjeru Mirko je napravio $N_U = 5$ promjena i $N_Q = 4$ upita.

Implementacija

Potrebno je priložiti datoteku koja implementira funkciju `init()` i `update()` te funkciju `calculate()` kao što je opisano ispod.

Predložci rješenja na vašem računalu (`game.c`, `game.cpp` i `game.pas`) sadrže implementaciju funkcije `gcd2(X, Y)` koja računa GCD vrijednost dva cijela nenegativna broja `X` and `Y`. Ako `X = Y = 0` onda `gcd2(X, Y)` vraća `0`.

Priložena funkcija je dovoljno brzo implementirana kako bi dobili sve bodove na ovom zadatku; preciznije, vrijeme izvršavanja funkcije proporcionalno je `log(X + Y)`.

Vaša funkcija: `init()`

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

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

Opis

Vaše rješenje mora implementirati ovu funkciju.

Pozivom ove funkcije vaše rješenje dobiva veličinu ploče i dopušta inicijalizaciju bilo kakvih globalnih varijabli i struktura podataka. Biti će pozvana samo jednom, prije bilo kojih poziva funkcija `update()` ili `calculate()`.

Parametri

- `R`: Broj redaka na ploči.
- `C`: Broj stupaca na ploči.

Vaša funkcija: `update()`

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

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

Opis

Vaše rješenje mora implementirati ovu funkciju.

Ova funkcija će biti pozvana kada Mirko promijeni vrijednost broja na nekom polju.

Parametri

- `P`: Redak polja koji se mijenja ($0 \leq P \leq R - 1$).
- `Q`: Stupac polja koji se mijenja ($0 \leq Q \leq C - 1$).
- `K`: Nova vrijednost koja će se postaviti na polje ($0 \leq K \leq 10^{18}$). Može biti ista kao i trenutna vrijednost.

Vaša funkcija: `calculate()`

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

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

Opis

Vaše rješenje mora implementirati ovu funkciju.

Ova funkcija mora izračunati najveću zajedničku mjeru svih brojeva u pravokutniku s nasuprotnim vrhovima u (P, Q) i (U, V) . Pravokutnik uključuje svoje bridove, tj. polja (P, Q) i (U, V) su uključena u izračun.

Ako su svi brojevi u pravokutniku jednaki nuli, traženo rješenje će također biti nula.

Parametri

- P : Redak gornjeg lijevog polja u pravokutniku ($0 \leq P \leq R - 1$).
- Q : Stupac gornjeg lijevog polja u pravokutniku ($0 \leq Q \leq C - 1$).
- U : Redak donjeg desnog polja u pravokutniku ($P \leq U \leq R - 1$).
- V : Stupac donjeg desnog polja u pravokutniku ($Q \leq V \leq C - 1$).
- *Povratna vrijednost*: Najveća zajednička mjera svih brojeva u pravokutniku ili 0 ako su svi promatrani brojevi nula.

Testni podaci

Prvi primjer:

Poziv	Rješenje
<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

Ograničenja

- Vremensko ograničenje: pogledati odjeljak o bodovanju.
- Memorijsko ograničenje: pogledati odjeljak o bodovanju.
- $1 \leq R, C \leq 10^9$
- $0 \leq K \leq 10^{18}$, gdje je `K` bilo koji broj koji je Mirko stavio na ploču.

Bodovanje

Pogledati službeni tekst zadatka na engleskom za informacije o bodovanju.

Lokalno testiranje

Grejder na vašem računalu će učitati ulazne podatke iz datoteke `game.in` sljedećeg oblika:

- redak 1: `R C N`
- sljedećih `N` redaka: jedan potez po liniji redoslijedom kojim su odigrani

Redak koji opisuje potez mora biti sljedećeg oblika:

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

Npr., prvi primjer bi bio predstavljen:

```
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
```

Napomene

C/C++ Potrebno je dodati `#include "game.h"`.

Pascal Potrebno je definirati `unit Game`. Svi nizovi moraju biti indeksirani od `0` (a ne od `1`).

Budući da brojevi na ploči mogu biti vrlo veliki preporučeno je koristiti `long long` tip podataka u C/C++ jeziku te `Int64` tip podataka u jeziku Pascal.