



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

game

Lithuanian – 1.1

Lentelėje yra R eilučių ir C stulpelių. Eilutės numeruojamos nuo 0 iki $R - 1$, o stulpeliai – nuo 0 iki $C - 1$. Langelis, esantis p -oje eilutėje ir q -ame stulpelyje, žymimas (p, q) . Pradžioje kiekviename langelyje įrašytas 0 .

Vienu žingsniu galima

- pakeisti langelyje (p, q) įrašytą skaičių;
- paklausti, koks yra stačiakampyje, kurio du priešingi kampai yra (p, q) ir (u, v) , esančių skaičių *didžiausias bendras daliklis* (BDD); šie kampai priklauso stačiakampiui.

Iš viso atliekama $N_U + N_Q$ žingsnių (N_U keitimų ir N_Q klausimų)

Jūsų užduotis – apskaičiuoti teisingus atsakymus į visus klausimus.

Pavyzdys

Tarkime, $R = 2$ ir $C = 3$, o pirmais žingsniais keičiami šie skaičiai:

- langelio $(0, 0)$ skaičius į 20 ;
- langelio $(0, 2)$ skaičius į 15 ;
- langelio $(1, 1)$ skaičius į 12 .

20	0	15
0	12	0

Po šių veiksmų lentelė atrodo kaip pavaizduota aukščiau. Tada užduodami klausimai, koks yra šių stačiakampių BDD:

- priešingi kampai $(0, 0)$ ir $(0, 2)$: trys skaičiai šiame stačiakampyje yra 20, 0 ir 15, o jų BDD yra 5;
- priešingi kampai $(0, 0)$ ir $(1, 1)$: keturi skaičiai šiame stačiakampyje yra 20, 0, 0 ir 12, o jų BDD yra 4.

Po to atliekami tokie pakeitimai:

- langelio $(0, 1)$ skaičius pakeičiamas į 6;
- langelio $(1, 1)$ skaičius pakeičiamas į 14.

20	6	15
0	14	0

Dabar lentelė atrodo kaip pavaizduota aukščiau. Užduodami klausimai, kokie yra šių stačiakampių BDD:

- priešingi kampai $(0, 0)$ ir $(0, 2)$: dabar trys skaičiai šiame stačiakampyje yra 20, 6 ir 15, o jų BDD yra 1;
- priešingi kampai $(0, 0)$ ir $(1, 1)$: dabar keturi skaičiai šiame stačiakampyje yra 20, 6, 0 ir 14, o jų BDD yra 2.

Buvo atlikti $N_U = 5$ pakeitimai ir užduoti $N_Q = 4$ klausimai.

Realizavimas

Pateikite failą, realizuojantį procedūras `init()` ir `update()` bei funkciją `calculate()`, kaip aprašyta žemiau.

Galite naudotis pavyzdiniuose sprendimuose (`game.c`, `game.cpp` ir `game.pas`) esančia funkcija `gcd2(X, Y)`, kuri apskaičiuoja dviejų neneigiamų sveikųjų skaičių X ir Y didžiausią bendrą daliklį. Jei $X = Y = 0$, tai `gcd2(X, Y)` grąžins `0`.

Naudodami šią funkciją galite surinkti visus taškus; jos ilgiausias veikimo laikas auga proporcingai $\log(X + Y)$.

Jūsų procedūra: `init()`

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

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

Veikimas

Realizuokite šią procedūrą.

Procedūra perduoda lentelės dydį ir leidžia jums inicializuoti globalius kintamuosius ir duomenų struktūras. Ji kviečiama tik vieną kartą prieš kreipinius į `update()` ir `calculate()`.

Parametrai

- `R` : eilučių skaičius.
- `C` : stulpelių skaičius.

Jūsų procedūra: `update()`

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

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

Veikimas

Realizuokite šią procedūrą.

Procedūra iškviečiama, kai į kažkurį langelį įrašomas skaičius.

Parametrai

- `P` : langelio ($0 \leq P \leq R - 1$) eilutė.
- `Q` : langelio ($0 \leq Q \leq C - 1$) stulpelis.
- `K` : naujai į langelį įrašomas skaičius ($0 \leq K \leq 10^{18}$); gali sutapti su buvusiu reikšme.

Jūsų funkcija: `calculate()`

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

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

Veikimas

Realizuokite šią funkciją.

Funkcija turi apskaičiuoti visų stačiakampyje esančių skaičių BDD. Stačiakampis nusakomas dviem priešingais kampais (P, Q) ir (U, V) . Langeliai (P, Q) ir (U, V) stačiakampiui priklauso.

Jei visi stačiakampyje įrašyti skaičiai lygūs 0, tuomet ši funkcija turi grąžinti 0.

Parametrai

- P : kairiojo viršutiniojo langelio eilutė ($0 \leq P \leq R - 1$).
- Q : kairiojo viršutiniojo langelio stulpelis ($0 \leq Q \leq C - 1$).
- U : dešiniojo apatinio langelio eilutė ($P \leq U \leq R - 1$).
- V : dešiniojo apatinio langelio stulpelis ($Q \leq V \leq C - 1$).
- *return*: visų stačiakampyje įrašytų sveikųjų skaičių BDD arba 0, jei visi tie skaičiai lygūs 0.

Pavyzdinis veikimas

Tokia sesija atitinka aprašytą pavyzdį:

Kreipinys į funkciją	Grąžinama
<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

Ribojimai

- Laiko ribojimas: žr. angliškas sąlygas
- Atminties ribojimas: žr. angliškas sąlygas
- $1 \leq R, C \leq 10^9$
- $0 \leq K \leq 10^{18}$; čia K yra sveikasis skaičius, naujai įrašomas į lentelę.

Dalinės užduotys

Dalinių užduočių ribojimai įrašyti angliškose sąlygose.

Subtask	Points	R	C	N_u	N_q	Time limit	Memory limit

Eksperimentavimas

Jūsų kompiuteryje įrašytas pavyzdinis vertintojas duomenis skaitys iš failo `game.in`, kurio formatas turi būti toks:

- 1-ma eilutė: `R C N`
- Tolesnės `N` eilučių: po vieną žingsnį eilutėje ta tvarka kuria jie vykdomi

Žingsniui skirta eilutė turi būti pateikta tokiu formatu:

- norint nurodyti `update(P, Q, K)` žingsnį: `1 P Q K`
- norint nurodyti `calculate(P, Q, U, V)` žingsnį: `2 P Q U V`

Aprašytas pavyzdys būtų pateiktas tokiu formatu:

```
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
```

Pastabos apie programavimo kalbas

C/C++ Įterpkite `#include "game.h"`.

Pascal Apibrėžkite `unit Game`. Visi masvai sunumeruoti pradedant nuo `0` (ne nuo `1`).

Lentelėje įrašyti sveikieji skaičiai gali būti labai dideli, todėl programuojantiems C/C++ rekomenduojame naudoti `long long` duomenų tipą, o programuojantiems Paskaliu rekomenduojama naudoti `Int64`.