International Olympiad in Informatics 2013



6-13 July 2013 Brisbane, Australia Day 2 tasks

game

Slovenščina — 1.0

Bazza in Shazza igrata igro na pravokotni mreži celic z R vrsticami, oštevilčenimi z 0, ..., R - 1, in C stolpci, oštevilčenimi z 0, ..., C - 1. Naj (P, Q) označuje celico v vrstici P in stolpcu Q. Vsaka celica vsebuje ne-negativno celo število. Na začetku igre so vsa števila enaka 0.

Oglejmo si, kako se igra odvija. Ob vsakem trenutku lahko Bazza izvede eno od sledečih dveh dejanj:

- posodobi število v celici (P, Q);
- prosi Shazzo, naj izračuna GCD (največji skupni delitelj) vseh števil znotraj pravokotnika, podanega z nasproti ležečima ogliščema (P, Q) in (U, V) (obe oglišči sta znotraj pravokotnika).

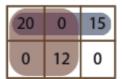
Bazza izvrši $N_U + N_Q$ dejanj (N_U posodobitev celic in N_Q povpraševanj po GCD), nato pa mu postane dolgčas in gre igrat kriket.

Tvoja naloga je določiti pravilne odgovore na Bazzova vprašanja.

Primer

Naj bo R = 2 in C = 3, Bazza pa prične igro s sledečimi dejanji:

- Posodobi celico (0, 0) na 20;
- Posodobi celico (0, 2) na 15;
- Posodobi celico (1, 1) na 12.

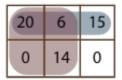


Nastala mreža je prikazana na zgornji sliki. Denimo, da Bazza zatem povpraša po GCD-jih za sledeče pravokotnike:

- Nasprotni oglišči (0, 0) in (0, 2): Ta pravokotnik vsebuje tri števila: 20, 0 in 15.
 Njihov GCD je 5.
- Nasprotni oglišči (0, 0) in (1, 1): Ta pravokotnik vsebuje štiri števila: 20, 0, 0 in 12.
 Njihov GCD je 4.

Recimo, da Bazza zatem izvrši sledeči posodobitvi:

- Posodobi celico (0, 1) na 6;
- Posodobi celico (1, 1) na 14.



Nastala mreža je prikazana na zgornji sliki. Denimo, da Bazza sedaj ponovno povpraša po GCD-jih sledečih pravokotnikov:

- Nasprotni oglišči (0, 0) in (0, 2): Ta pravokotnik sedaj vsebuje števila 20, 6 in 15, njihov GCD pa je 1.
- Nasprotni oglišči (0, 0) in (1, 1): Ta pravokotnik sedaj vsebuje števila 20, 6, 0 in 14, njihov GCD pa je 2.

Bazza je v prikazanem primeru izvršil $N_U = 5$ posodobitev in $N_Q = 4$ povpraševanj.

Implementacija

Oddaj datoteko, v kateri sta implementirani proceduri (init ()) in (update ()) ter funkcija (calculate ()) po sledečih navodilih:

Predloge rešitev na tvojem računalniku (game.c), game.cpp in game.pas) že vsebujejo funkcijo [gcd2 (X, Y)], ki izračuna največji skupni delitelj podanih dveh ne-negativnih celih števil X in Y. Če je X = Y = 0, potem [gcd2 (X, Y)] prav tako vrne [0].

Ta funkcija je dovolj hitra, da z njo lahko dosežeš polno število točk. Deluje v času, ki je v najslabšem primeru premosorazmeren log(X + Y).

Tvoja procedura: init()

```
C/C++ void init(int R, int C);
Pascal procedure init(R, C : LongInt);
```

Opis

Ta procedura sprejme velikost mreže in ti omogoča inicializacijo in katere koli globalne spremenljivke in podatkovne strukture. Klicana bo samo enkrat, pred katerem koli klicu update() in calculate().

Parametri

- R: Število vrstic.
- C: Število stolpcev.

Tvoja procedura: update()

```
C/C++ void update(int P, int Q, long long K);
Pascal procedure update(P, Q : LongInt; K : Int64);
```

Opis

Procedura se kliče, ko Bazza posodobi število v neki celici.

Parametri

- P: Vrstica celice $(0 \le P \le R 1)$.
- Q: Stolpec celice $(0 \le Q \le C 1)$.
- K: Novo število za celico (0 ≤ K ≤ 10¹⁸). Pozor: to število je lahko enako obstoječemu.

Tvoja 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

Ta funkcija naj izračuna največji skupni delitelj vseh števil v pravokotniku z nasprotnima ogliščema (P, Q) in (U, V). Celici (P, Q) in (U, V) sta znotraj pravokotnika.

Če so vsa števila pravokotnika ničle, potem naj tudi funkcija vrne ničlo.

Parametri

- P: Vrstica zgornje leve celice pravokotnika $(0 \le P \le R 1)$.
- Q: Stolpec zgornje leve celice pravokotnika ($0 \le Q \le C 1$).
- U: Vrstica spodnje desne celice pravokotnika ($P \le U \le R 1$).
- V: Stolpec spodnje desne celice pravokotnika ($Q \le V \le C 1$).
- *Vrača*: GCD vseh števil v pravokotniku oziroma 0, če so vsa ta števila ničle.

Vzorčno zaporedje dejanj

Sledeči klici ustrezajo zgornjemu primeru:

Klic	Vrača	
init(2, 3)		
update(0, 0, 2	0)	
update(0, 2, 1	5)	
update(1, 1, 1	2)	
calculate(0, 0	, 0, 2)	5
calculate(0, 0	, 1, 1)	4
update(0, 1, 6)	
update(1, 1, 1	4)	
calculate(0, 0	, 0, 2)	1
calculate(0, 0	, 1, 1)	2

Omejitve

- Časovna omejitev: podana v podnalogah
- Prostorska omejitev: podana v podnalogah
- $1 \le R, C \le 10^9$
- $0 \le K \le 10^{18}$, kjer je K katerokoli število, ki ga Bazza vpiše v celico.

Podnaloge

Podnaloga	Točke	R	С	N _U	N _Q	Časovna omejitev	Prostorska omejitev

Preizkušanje

Vzorčni ocenjevalnik na tvojem računalniku bere vhod iz datoteke game.in, ki mora biti sledeče oblike:

- vrstica 1: R C N
- naslednjih N vrstic: ena vrstica za vsako dejanje. Vrstni red vrstic ustreza vrstnemu redu dogodkov.

Vsaka vrstica posameznega dejanja mora biti sledeče oblike:

```
■ za dejanje update(P, Q, K): 1 P Q K
```

■ za dejanje calculate(P, Q, U, V): 2 P Q U V

Zgornji primer bi bil podan v sledeči obliki:

```
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
```

Jezikovne opombe

```
C/C++ Potrebuješ #include "game.h".

Pascal Definiraj unit Game. Oštevilčenje vseh polj se prične z 0 in ne z 1.
```

Ker so števila v celicah lahko zelo velika, uporabnikom C/C++ priporočamo uporabo tipa long long, uporabnikom Pascala pa Int64.