International Olympiad in Informatics 2013



6-13 July 2013 Brisbane, Australia Day 2 tasks

game

Italian -1.1

Bazza e Shazza iniziano un gioco. Il gioco si svolge su una griglia di celle, con R righe numerate 0, ..., R - 1, e C colonne numerate 0, ..., C - 1 (useremo (P, Q) per indicare la cella nella riga P e nella colonna Q). Ogni cella contiene un intero non negativo, che all'inizio del gioco è zero.

Il gioco si svolge come segue. In qualunque momento, Bazza può:

- aggiornare il valore di una cella (P, Q), assegnandole un intero non negativo (update);
- chiedere a Shazza di calcolare il massimo comune divisore (GCD) di tutti gli interi in un certo blocco rettangolare di celle, racchiuso tra gli angoli opposti (P, Q) e (U, V) inclusi (*query*).

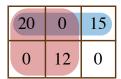
Bazza può fare al massimo $|N_U + N_Q|$ azioni in totale ($|N_U|$ update e $|N_Q|$ query) prima di stufarsi e andare fuori a giocare a cricket.

Il tuo compito è di trovare le risposte corrette per Shazza.

Esempio

Data una griglia di dimensioni R = 2 e C = 3, Bazza inizia con i seguenti aggiornamenti:

- Aggiorna la cella (0, 0) a 20;
- Aggiorna la cella (0, 2) a 15;
- Aggiorna la cella (1, 1) a 12.

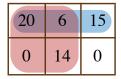


La griglia risultante è mostrata nell'immagine soprastante. Bazza richiede quindi il GCD dei seguenti rettangoli di celle:

- Rettangolo con angoli (0, 0) e (0, 2): i tre interi in questo rettangolo sono 20, 0 e 15, e il loro GCD è 5.
- Rettangolo con angoli (0, 0) e (1, 1): i quattro interi in questo rettangolo sono 20, 0, 0, 12 e il loro GCD è 4.

Ora Bazza esegue i seguenti aggiornamenti:

- Aggiorna la cella (0, 1) a 6;
- Aggiorna la cella (1, 1) a 14.



La nuova griglia è riportata nell'immagine soprastante. Bazza ora richiede il GCD dei seguenti rettangoli:

- Rettangolo con angoli (0, 0) e (0, 2): ora i tre interi nel rettangolo sono 20, 6 e 15, e il loro GCD è 1.
- Rettangolo con angoli (0, 0) e (1, 1): ora i quattro interi nel rettangolo sono 20, 6, 0 e 14, e il loro GCD è 2.

In questo caso Bazza ha eseguito $N_U = 5$ update e $N_Q = 4$ query, per un totale di nove azioni.

Implementazione

Devi sottoporre un file che implementi le procedure <code>init()</code> e <code>update()</code> e la funzione <code>calculate()</code>, come segue.

Per aiutarti, i template delle soluzioni fornite sul computer (game.c, game.cpp) e game.pas) includono la funzione (gcd2 (X, Y)) che calcola il massimo comune divisore tra due numeri interi non negativi X e Y. Nel caso (X = Y = 0) la funzione (gcd2 (X, Y)) restituirà (0).

Questa funzione è sufficientemente veloce per ottenere punteggio pieno; in particolare, il tempo di esecuzione è proporzionale a log(X + Y) nel caso peggiore.

Procedura init()

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

Descrizione

La tua sottoposizione deve implementare questa procedura.

Questa procedura riceve la dimensione iniziale della griglia e ti permette di inizializzare variabili globali e strutture di dati. È chiamata una sola volta, prima delle chiamate a update() e calculate().

Parametri

- R: Il numero di righe.
- C: Il numero di colonne.

Procedura: update()

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

Descrizione

La tua sottoposizione deve implementare questa procedura.

Questa procedura verrà chiamata ogni volta che Bazza aggiorna il numero in una cella della griglia.

Parametri

- P: La riga della cella $(0 \le P \le R 1)$.
- Q: La colonna della cella $(0 \le Q \le C 1)$.
- K: Il nuovo intero nella cella ($0 \le K \le 10^{18}$).

Funzione: calculate()

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

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

Descrizione

La tua sottoposizione deve implementare questa funzione.

Questa funzione deve calcolare il GCD tra tutti gli interi del rettangolo di vertici opposti (P, Q) e (U, V). Gli estremi sono inclusi, cioè le celle (P, Q) e (U, V) fanno parte del rettangolo.

Se tutti gli interi in questo rettangolo sono zero, la funzione deve restituire zero.

Parametri

- P: La riga della cella in alto a sinistra del rettangolo $(0 \le P \le R 1)$.
- \mathbb{Q} : La colonna della cella in alto a sinistra del rettangolo ($0 \le \mathbb{Q} \le \mathbb{C} 1$).
- U: La riga della cella in basso a destra del rettangolo ($P \le U \le R 1$).
- V: La colonna della cella in basso a destra del rettangolo ($Q \le V \le C 1$).
- *Restituisce*: Il GCD tra tutti gli interi del rettangolo, oppure 0 se tutti gli interi nel rettangolo sono zero.

Sessione di esempio

La seguente sessione descrive l'esempio riportato sopra.

Chiamata a funzione		Restituisce
init(2, 3)		
update(0, 0,	20)	
update(0, 2,	15)	
update(1, 1,	12)	
calculate(0,	0, 0, 2)	5
calculate(0,	0, 1, 1)	4
update(0, 1,	6)	
update(1, 1,	14)	
calculate(0,	0, 0, 2)	1
calculate(0,	0, 1, 1)	2

Limiti

- Tempo limite: vedi subtask
- Limite di memoria: vedi subtask
- $1 \le R, C \le 10^9$
- $0 \le K \le 10^{18}$, dove K è un qualunque numero intero che Bazza inserisce in una cella della griglia.

Subtask

Controllare la versione in inglese per i parametri dei subtask.

Testing

Il grader di esempio legge l'input dal file game.in, che deve essere nel seguente formato:

- linea 1: [R C N
- le successive N linee: una azione per linea, nell'ordine in cui avvengono.

La linea per ciascuna azione deve essere in uno dei seguenti formati:

```
per l'azione update (P, Q, K): 1 P Q K
per l'azione calculate (P, Q, U, V): 2 P Q U V
```

L'esempio riportato sopra può essere fornito nel seguente formato:

```
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 relative al linguaggio

```
C/C++ Devi inserire #include "game.h".

Pascal Devi definire unit Game. Tutti gli array sono numerati a partire da 0 (e non 1).
```

Gli interi nelle celle della griglia possono essere molto grandi: si consiglia di utilizzare il tipo di dato long long in C/C++ e il tipo di dato Int64 in Pascal.