



International Olympiad in Informatics 2013

6-13 July 2013
Brisbane, Australia
Day 2 tasks

game

Deutsch — 1.1

Bazza und Shazza spielen ein Spiel, auf einem Spielbrett mit R Zeilen, nummeriert mit $0, \dots, R - 1$ und C Spalten, nummeriert mit $0, \dots, C - 1$. Im folgenden bezeichnet (P, Q) die Zelle in Zeile P und Spalte Q . Jede Zelle enthält eine nicht negative ganze Zahl, und zu Beginn des Spiels sind alle diese Zahlen null.

Das Spiel geht wie folgt: Zu jedem Zeitpunkt kann Bazza

- entweder eine Zelle (P, Q) aktualisieren, indem er ihr die ganze Zahl zuweist, die sie enthalten soll;
- oder Shazza auffordern, den grössten gemeinsamen Teiler (ggT) aller Zahlen innerhalb eines rechteckigen Blocks von Zellen zu berechnen, inklusive den gegenüberliegenden Ecken (P, Q) und (U, V) . (Der ggT heisst auf Englisch GCD, was für „greatest common divisor“ steht.)

Bazza wird $N_U + N_Q$ Aktionen ausführen: N_U mal aktualisiert (update) er eine Zelle und N_Q (question) mal fordert er Shazza auf, einen ggT zu berechnen. Dann beginnt er sich zu langweilen und geht Cricket spielen.

Deine Aufgabe ist es, die korrekten Antworten auszurechnen.

Beispiel

Angenommen, $R = 2$ und $C = 3$, und Bazza beginnt mit den folgenden Aktualisierungen:

- Zelle $(0, 0)$ zu 20 aktualisieren;
- Zelle $(0, 2)$ zu 15 aktualisieren;
- Zelle $(1, 1)$ zu 12 aktualisieren.

| | | |
|----|----|----|
| 20 | 0 | 15 |
| 0 | 12 | 0 |

Obiges Bild zeigt das resultierende Spielbrett. Bazza könnte nun die ggTs der folgenden Rechtecke abfragen:

- Gegenüberliegende Ecken $(0, 0)$ und $(0, 2)$: Die drei Zahlen in diesem Rechteck sind 20, 0 und 15, und ihr ggT ist 5.
- Gegenüberliegende Ecken $(0, 0)$ und $(1, 1)$: Die vier Zahlen in diesem Rechteck sind 20, 0, 0 und 12, und ihr ggT ist 4.

Nun macht Bazza folgende Aktualisierungen:

- Zelle $(0, 1)$ zu 6 aktualisieren;
- Zelle $(1, 1)$ zu 14 aktualisieren.

| | | |
|----|----|----|
| 20 | 6 | 15 |
| 0 | 14 | 0 |

Obiges Bild zeigt das resultierende Spielbrett. Bazza könnte nun die ggTs der folgenden Rechtecke abfragen:

- Gegenüberliegende Ecken $(0, 0)$ und $(0, 2)$: Nun sind die drei Zahlen in diesem Rechteck 20, 6 und 15, und ihr ggT ist 1.
- Gegenüberliegende Ecken $(0, 0)$ und $(1, 1)$: Nun sind die vier Zahlen in diesem Rechteck 20, 6, 0 und 14, und ihr ggT ist 2.

Hier hat Bazza also $N_U = 5$ Aktualisierungen gemacht und $N_Q = 4$ Fragen gestellt.

Implementierung

Du sollst eine Datei einsenden, welche die Prozeduren `init()` und `update()` und die Funktion `calculate()` wie unten beschrieben implementiert.

Um dir zu helfen, enthalten auf deinem Computer die Beispieldateien (`game.c`, `game.cpp` und `game.pas`) jeweils die Funktion `gcd2(X, Y)`, welche den grössten gemeinsamen Teiler von zwei gegebenen nicht negativen Ganzzahlen X und Y berechnet. Wenn $X = Y = 0$, dann gibt `gcd2(X, Y)` auch 0 zurück.

Diese Funktion ist schnell genug um alle Punkte zu erreichen; genauer gesagt ist die Laufzeit im schlechtesten Fall proportional zu `log(X + Y)`.

Deine Prozedur: `init()`

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

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

Beschreibung

Deine Abgabe muss diese Prozedur implementieren.

Diese Prozedur teilt dir die Grösse des Feldes mit und ermöglicht dir, globale Variablen und Datenstrukturen zu initialisieren. Diese Prozedur wird nur einmal aufgerufen, nämlich vor sämtlichen Aufrufen von `update()` oder `calculate()`.

Parameter

- `R` : Die Anzahl der Zeilen.
- `C` : Die Anzahl der Spalten.

Deine Prozedur: `update()`

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

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

Beschreibung

Deine Abgabe muss diese Prozedur implementieren.

Diese Prozedur wird aufgerufen, wenn Bazza einer Zelle eine Zahl zuweist.

Parameter

- `P` : Die Zeilennummer der Zelle ($0 \leq P \leq R - 1$).
- `Q` : Die Spaltennummer der Zelle ($0 \leq Q \leq C - 1$).
- `K` : Die neue ganze Zahl für diese Zelle ($0 \leq K \leq 10^{18}$). Diese ist möglicherweise identisch mit der aktuellen Zahl.

Deine Funktion: `calculate()`

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

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

Beschreibung

Deine Abgabe muss diese Funktion implementieren.

Diese Funktion soll den grössten gemeinsamen Teiler der Zahlen im Rechteck mit den gegenüberliegenden Ecken `(P, Q)` und `(U, V)` berechnen. Dabei gehören die Zellen `(P, Q)` und `(U, V)` auch zum Rechteck.

Wenn alle Zahlen in diesem Rechteck `0` sind, dann soll diese Funktion `0` zurückgeben.

Parameter

- `P` : Die Zeilennummer der linken oberen Zelle des Rechtecks ($0 \leq P \leq R - 1$).
- `Q` : Die Spaltennummer der linken oberen Zelle des Rechtecks ($0 \leq Q \leq C - 1$).
- `U` : Die Zeilennummer der rechten unteren Zelle des Rechtecks ($P \leq U \leq R - 1$).
- `V` : Die Spaltennummer der rechten unteren Zelle des Rechtecks ($Q \leq V \leq C - 1$).
- *Rückgabe*: Der ggT aller Zahlen im Rechteck bzw. `0`, falls alle diese Zahlen 0 sind.

Beispiel-Session

Folgender Ablauf beschreibt das obige Beispiel:

| Funktionsaufruf | Rückgabe |
|------------------------------------|----------|
| <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 |

Beschränkungen

- Zeitlimit: Siehe Teilaufgaben
- Speicherlimit: Siehe Teilaufgaben
- $1 \leq R, C \leq 10^9$
- $0 \leq K \leq 10^{18}$ für alle ganzen Zahlen K , die Bazza in eine Zelle schreibt.

Teilaufgaben

Siehe englische Version für die Beschränkungen und Punktzahlen der Teilaufgaben.

| Teilaufgabe | Punkte | R | C | N_u | N_q | Zeitlimit | Speicherlimit |
|-------------|--------|---|---|-------|-------|-----------|---------------|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Testen

Der Beispielgrader auf deinem Computer liest die Eingabe aus der Datei `game.in`, die wie folgt aufgebaut ist:

- Zeile 1: `R C N`
- nächste `N` Zeilen: Eine Aktion pro Zeile, in der Reihenfolge, in der sie auftreten.

Die Zeile für jede Aktion muss eines der folgenden Formate besitzen:

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

Das gegebene Beispiel müsste folgendes Format besitzen:

```
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
```

Sprachspezifische Bemerkungen

C/C++ Du musst `#include "game.h"` verwenden.

Pascal Du musst `unit Game` definieren. Alle Arrays sind mit `0` beginnend indiziert (nicht `1`).

Weil die Zahlen in den Zellen sehr gross sein können, sollten C/C++ User den Datentyp `long long` und Pascal User den Datentyp `Int64` verwenden.