

Mozaik

Selma planira naslikati mozaik na zidu. Mozaik predstavljamo kao $N \times N$ tabelu koja je načinjena od N^2 jediničnih 1×1 polja koja su u početku prazna (neobojena). Redovi mozaika označeni su brojevima od 0 do $N - 1$, gledajući odozgo prema dolje. Kolone su na sličan način označene brojevima od 0 do $N - 1$, s lijeva na desno. Polje u redu i i koloni j ($0 \leq i < N$, $0 \leq j < N$), označeno je s (i, j) . Svako polje mora biti obojeno u bijelo (označeno s 0) ili crno (označeno 1).

Kako bi obojila mozaik, Selma prvo odabire nizove X i Y dužine N . Nizovi se sastoje od vrijednosti 0 i 1, sa dodatnim ograničenjem da je $X[0] = Y[0]$. Selma boji prvi red odozgo (red 0) prema nizu X , tačnije boji polje $(0, j)$ u boju $X[j]$ ($0 \leq j < N$). Također na sličan način boji prvu kolonu s lijeva (kolona 0) prema nizu Y , tako da boja polja $(i, 0)$ bude $Y[i]$ ($0 \leq i < N$).

Selma nakon toga ponavlja sljedeću proceduru sve dok sva polja nisu obojena:

- Pronalazi *neobojeno* polje (i, j) takvo da su gornji susjed (polje $(i - 1, j)$) i lijevi susjed (polje $(i, j - 1)$) oba *već obojeni*.
- Nakon toga ona boji polje (i, j) u crno ako su oba susjeda bijele boje; u suprotnom, polje (i, j) boji u bijelo.

Može se pokazati da finalna konfiguracija boja ne ovisi o redoslijedu kojim Selma boji polja.

Jasmin je jako znatiželjan o bojama u mozaiku. On će postaviti Selmi Q pitanja, označena od 0 do $Q - 1$. U pitanju k ($0 \leq k < Q$), Jasmin opisuje pravougaonik u mozaiku na sljedeći način:

- Prvi red odozgo redak pravougaonika je $T[k]$, a posljednji red odozgo $B[k]$ ($0 \leq T[k] \leq B[k] < N$),
- Prva kolona s lijeva pravougaonika je $L[k]$, a posljednja kolona s lijeva $R[k]$ ($0 \leq L[k] \leq R[k] < N$).

Odgovor na pitanje je broj crnih polja unutar zadanog pravougaonika. Tačnije, Selma treba saznati koliko polja (i, j) postoji, takvih da $T[k] \leq i \leq B[k]$, $L[k] \leq j \leq R[k]$ te da je polje (i, j) obojeno u crno.

Napišite program koji odgovara na jasminova pitanja.

Implementacijski detalji

Potrebno je implementirati sljedeće procedure.

```
std::vector<long long> mosaic(
    std::vector<int> X, std::vector<int> Y,
    std::vector<int> T, std::vector<int> B,
    std::vector<int> L, std::vector<int> R)
```

- X, Y : nizovi dužine N opisuju boje polja u prvom redu odozgo, te prvom redu s lijeva, redom.
- T, B, L, R : nizovi dužine Q opisuju pravougaonike iz Jasminovih pitanja.
- Procedura treba vratiti niz C dužine Q , takav da $C[k]$ daje odgovor na pitanje k ($0 \leq k < Q$).
- Procedura će biti pozvana tačno jednom po testnom primjeru.

Ograničenja

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 200\,000$
- $X[i] \in \{0, 1\}$ i $Y[i] \in \{0, 1\}$ za svaki i takav da $0 \leq i < N$
- $X[0] = Y[0]$
- $0 \leq T[k] \leq B[k] < N$ i $0 \leq L[k] \leq R[k] < N$ za svaki k takav da $0 \leq k < Q$

Podzadaci

Podzadatak	Bodovi	Dodatna ograničenja
1	5	$N \leq 2; Q \leq 10$
2	7	$N \leq 200; Q \leq 200$
3	7	$T[k] = B[k] = 0$ (za svaki k takav da $0 \leq k < Q$)
4	10	$N \leq 5000$
5	8	$X[i] = Y[i] = 0$ (za svaki i takav da $0 \leq i < N$)
6	22	$T[k] = B[k]$ and $L[k] = R[k]$ (za svaki k takav da $0 \leq k < Q$)
7	19	$T[k] = B[k]$ (za svaki k takav da $0 \leq k < Q$)
8	22	Bez dodatnih ograničenja.

Primjer

Razmotrite sljedeći poziv procedure.

```
mosaic([1, 0, 1, 0], [1, 1, 0, 1], [0, 2], [3, 3], [0, 0], [3, 2])
```

Ovaj primjer je ilustriran u slikama ispod. Lijeva slika prikazuje boje u mozaiku. Srednja i desna slika prikazuju pravougaonike iz Jasminovog prvog i drugog pitanja, redom.

	0	1	2	3
0	1	0	1	0
1	1	0	0	1
2	0	1	0	0
3	1	0	1	0

	0	1	2	3
0	1	0	1	0
1	1	0	0	1
2	0	1	0	0
3	1	0	1	0

	0	1	2	3
0	1	0	1	0
1	1	0	0	1
2	0	1	0	0
3	1	0	1	0

Odgovori na pitanja (tj., broj jedinica u osjenčenim pravougaonicima) su 7, odnosno 3. Dakle, procedura treba vratiti $[7, 3]$.

Sample Grader

Ulazni format:

```
N
X[0] X[1] ... X[N-1]
Y[0] Y[1] ... Y[N-1]
Q
T[0] B[0] L[0] R[0]
T[1] B[1] L[1] R[1]
...
T[Q-1] B[Q-1] L[Q-1] R[Q-1]
```

Izlazni format:

```
C[0]
C[1]
...
C[S-1]
```

Ovdje S predstavlja dužinu niza C , koju vraća mozaic.