

Mosaic

Lavigna vuole colorare un mosaico di argilla su una parete. Il mosaico è una griglia $N \times N$, costituita da N^2 tessere quadrate 1×1 inizialmente non colorate. Le righe del mosaico sono numerate da 0 a $N - 1$ dall'alto verso il basso, e le colonne sono numerate da 0 a $N - 1$ da sinistra a destra. La tessera nella riga i e nella colonna j ($0 \leq i < N$, $0 \leq j < N$) è indicata con (i, j) . Ogni tessera deve essere colorata di bianco (indicato con 0) o nero (indicato con 1).

Per colorare il mosaico, Lavigna sceglie prima due array X e Y di lunghezza N , ciascuno costituito dai valori 0 e 1, tali che $X[0] = Y[0]$. Colora quindi le tessere della riga 0 (la più in alto) secondo la matrice X , cioè in modo che il colore della tessera $(0, j)$ sia $X[j]$ ($0 \leq j < N$); e le tessere della colonna 0 (la più a sinistra) secondo la matrice Y , cioè in modo che il colore della tessera $(i, 0)$ sia $Y[i]$ ($0 \leq i < N$).

Poi ripete i seguenti passaggi fino a quando tutte le tessere sono colorate:

- Trova una qualsiasi tessera *non colorata* (i, j) tale che il suo vicino in alto (tessera $(i - 1, j)$) e il vicino a sinistra (tessera $(i, j - 1)$) sono entrambi *già colorati*.
- Quindi, colora la tessera (i, j) di nero se entrambi i vicini sono bianchi; altrimenti colora la tessera (i, j) di bianco.

Si può dimostrare che i colori finali delle piastrelle non dipendono dall'ordine in cui Lavigna le sta colorando.

Yasmin vorrebbe conoscere i colori delle tessere del mosaico, quindi pone a Lavigna Q domande, numerate da 0 a $Q - 1$. Nella domanda k ($0 \leq k < Q$), Yasmin specifica un sottorettangolo del mosaico tramite:

- la riga più in alto $T[k]$ e la riga più in basso $B[k]$ ($0 \leq T[k] \leq B[k] < N$),
- la colonna più a sinistra $L[k]$ e la colonna più a destra $R[k]$ ($0 \leq L[k] \leq R[k] < N$),

e vuole sapere quante tessere nere sono contenute in quel sottorettangolo.

Nello specifico, Lavigna deve calcolare quante tessere (i, j) soddisfano entrambe le seguenti condizioni:

- $T[k] \leq i \leq B[k]$, $L[k] \leq j \leq R[k]$;
- il colore della tessera (i, j) è nero.

Scrivi un programma che risponda alle domande di Yasmin.

Note di implementazione

Devi implementare la seguente funzione.

```
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 : array di lunghezza N che descrivono i colori delle tessere rispettivamente nella riga più in alto e nella colonna più a sinistra.
- T, B, L, R : array di lunghezza Q che descrivono le domande poste da Yasmin.
- La funzione deve restituire un array C di lunghezza Q , tale che $C[k]$ è la risposta alla domanda k ($0 \leq k < Q$).
- Questa funzione viene chiamata esattamente una volta per ogni caso di test.

Assunzioni

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 200\,000$
- $X[i] \in \{0, 1\}$ e $Y[i] \in \{0, 1\}$ per ogni i tale che $0 \leq i < N$
- $X[0] = Y[0]$
- $0 \leq T[k] \leq B[k] < N$ e $0 \leq L[k] \leq R[k] < N$ per ogni k tale che $0 \leq k < Q$

Subtask

Subtask	Punteggio	Limitazioni aggiuntive
1	5	$N \leq 2; Q \leq 10$.
2	7	$N \leq 200; Q \leq 200$.
3	7	$T[k] = B[k] = 0$ (per ogni $0 \leq k < Q$).
4	10	$N \leq 5000$.
5	8	$X[i] = Y[i] = 0$ (per ogni $0 \leq i < N$).
6	22	$T[k] = B[k]$ e $L[k] = R[k]$ (per ogni $0 \leq k < Q$).
7	19	$T[k] = B[k]$ (per ogni $0 \leq k < Q$).
8	22	Nessuna limitazione aggiuntiva.

Esempio

Consideriamo la seguente chiamata.

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

Questo esempio è illustrato nelle immagini sottostanti. L'immagine a sinistra mostra i colori delle tessere del mosaico. Le immagini al centro e a destra mostrano i sottorettangoli che Yasmin ha chiesto rispettivamente nella prima e nella seconda domanda.

	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

Le risposte alle domande (cioè il numero di uni nei rettangoli ombreggiati) sono rispettivamente 7 e 3. Quindi, la funzione deve restituire [7, 3].

Grader di esempio

Formato di input:

```
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]
```

Formato di output:

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

dove S è la lunghezza dell'array C restituito da `mosaic`.