

Mosaïque

Salma prévoit de mettre en couleur une mosaïque murale. La mosaïque est une grille $N \times N$, composée de N^2 carreaux 1×1 initialement non-peints. Les lignes de la mosaïque sont numérotées de 0 à $N - 1$ de haut en bas et les colonnes sont numérotées de 0 à $N - 1$ de gauche à droite. Le carreau dans la ligne i et colonne j ($0 \leq i < N$, $0 \leq j < N$) est symbolisé (i, j) . Chaque carreau doit être peint soit en blanc (représenté par "0"), soit en noir (représenté par "1").

Pour peindre la mosaïque, Salma commence par prendre 2 tableaux X et Y de longueur N , chacun composés de valeurs 0 et 1 et dont $X[0] = Y[0]$. Elle peint les carreaux sur la première ligne (ligne 0) suivant le tableau X pour que la couleur du carreau $(0, j)$ soit $X[j]$ ($0 \leq j < N$). Elle peint également les carreaux de la première colonne (colonne 0) suivant le tableau Y , pour que la couleur du carreau $(i, 0)$ soit $Y[i]$ ($0 \leq i < N$).

Ensuite, elle répète les étapes suivantes jusqu'à ce que tous les carreaux soient peints :

- Elle trouve un carreau *non-peint* (i, j) dont le voisin du dessus $(i - 1, j)$ et le voisin gauche $(i, j - 1)$ sont tous deux *déjà peints*.
- Ensuite, elle peint (i, j) en noir si ces deux voisins sont blancs, en noir sinon.

Il peut être montré que le résultat final ne dépend pas de l'ordre dans lequel Salma peint les carreaux.

Yasmin s'intéresse aux couleurs des carreaux de la mosaïque. Elle pose à Salma Q questions, numérotées de 0 à $Q - 1$. Dans la question k ($0 \leq k < Q$), Yasmin spécifie un sous-rectangle de la mosaïque par :

- sa ligne supérieure $T[k]$ et sa ligne inférieure $B[k]$ ($0 \leq T[k] \leq B[k] < N$),
- sa colonne la plus à gauche $L[k]$ et sa colonne la plus à droite $R[k]$ ($0 \leq L[k] \leq R[k] < N$).

La réponse à la question est le nombre de carreaux noirs dans ce sous-rectangle. Autrement dit, Salma doit trouver combien de carreaux (i, j) existent parmi $T[k] \leq i \leq B[k]$, $L[k] \leq j \leq R[k]$ dont la couleur est noire.

Ecrivez un programme qui répond aux questions de Yasmin.

Détails d'implémentation

Vous devez implémenter la fonction suivante :

```
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 : tableaux de longueur N décrivant respectivement la couleur des carreaux de la ligne supérieure et de celle la plus à gauche.
- T, B, L, R : tableaux de longueur Q décrivant les questions de Yasmin
- La fonction doit renvoyer un tableau C de longueur Q dont les valeurs $C[k]$ répondent à la question k ($0 \leq k < Q$).
- Cette fonction est appelée une seule fois pour chaque test.

Contraintes

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 200\,000$
- $X[i] \in \{0, 1\}$ et $Y[i] \in \{0, 1\}$ pour tout i tel que $0 \leq i < N$
- $X[0] = Y[0]$
- $0 \leq T[k] \leq B[k] < N$ et $0 \leq L[k] \leq R[k] < N$ pour tout k tel que $0 \leq k < Q$

Sous-tâches

Sous-tâche	Score	Contraintes supplémentaires
1	5	$N \leq 2; Q \leq 10$
2	7	$N \leq 200; Q \leq 200$
3	7	$T[k] = B[k] = 0$ (pour tout k tel que $0 \leq k < Q$)
4	10	$N \leq 5000$
5	8	$X[i] = Y[i] = 0$ (pour tout i tel que $0 \leq i < N$)
6	22	$T[k] = B[k]$ et $L[k] = R[k]$ (pour tout k tel que $0 \leq k < Q$)
7	19	$T[k] = B[k]$ (pour tout k tel que $0 \leq k < Q$)
8	22	Aucune contrainte supplémentaire

Exemple

Considérez l'appel suivant :

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

Cet exemple est illustré dans les images ci-dessous. L'image de gauche montre les couleurs des carreaux. L'image du milieu et de droite montrent les sous-rectangles que Yasmin a posé dans sa première et seconde questions.

	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

Les réponses aux questions (c'est-à-dire le nombre de uns dans les parties grisées) sont respectivement de 7 et 3. La fonction doit donc renvoyer $[7, 3]$.

Évaluateur d'exemple (grader)

Format d'entrée :

```
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]
```

Format de sortie :

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

où S est la longueur du tableau C renvoyé par `mosaic`.