

## მოზაიკა

საღმა კედელზე თიხის მოზაიკის გაფერადებას გეგმავს. მოზაიკა არის  $N \times N$  ცხრილი, შემდგარი თავდაპირველად უფერო  $N^2$  რაოდენობის  $1 \times 1$  კვადრატული ფილებით. მოზაიკის სტრიქონები დანომრილია 0-დან  $(N - 1)$ -მდე ზემოდან ქვემოთ და სვეტები დანომრილია 0-დან  $(N - 1)$ -მდე მარცხნიდან მარჯვნივ. ფილა სტრიქონში  $i$  და სვეტში  $j$  ( $0 \leq i < N$ ,  $0 \leq j < N$ ) აღნიშნება  $(i, j)$ -ით. თითოეული ფილა უნდა იყოს თეთრი ფერის (აღნიშნულია 0-ით), ან შავი ფერის (აღნიშნულია 1-ით).

მოზაიკის გასაფერადებლად საღმა ჯერ არჩევს  $N$  სიგრძის ორ  $X$  და  $Y$  მასივს, თითოეულს შემდგარს 0 და 1 მნიშვნელობებისაგან, ისეთს, რომ  $X[0] = Y[0]$ . იგი აფერადებს ზედა სტრიქონის (სტრიქონი 0) ფილებს  $X$  მასივის მიხედვით ისე, რომ  $(0, j)$  ფილის ფერი არის  $X[j]$  ( $0 \leq j < N$ ). ის ასევე აფერადებს მარცხენა სვეტის ფილებს (სვეტი 0)  $Y$  მასივის მიხედვით ისე, რომ  $(i, 0)$  ფილის ფერი არის  $Y[i]$  ( $0 \leq i < N$ ).

შემდეგ ის იმეორებს შემდეგ ნაბიჯებს, სანამ ყველა ფილა არ გაფერადდება:

- ის პოულობს ნებისმიერ ისეთ *გაუფერადებელ*  $(i, j)$  ფილას, რომ მისი ზედა მეზობელი  $((i - 1, j)$  ფილა) და მარცხენა მეზობელი  $((i, j - 1)$  ფილა) ორივე არის უკვე გაფერადებული.
- შემდეგ იგი აფერადებს  $(i, j)$  ფილას შავად, თუ ორივე მეზობელი თეთრია. წინააღმდეგ შემთხვევაში იგი აფერადებს  $(i, j)$  ფილას თეთრად.

შეიძლება ვაჩვენოთ, რომ ფილების საბოლოო ფერები არ არის დამოკიდებული იმაზე, თუ რა თანმიმდევრობით აფერადებს მათ საღმა.

იასმინს ძალიან აინტერესებს მოზაიკის ფილის ფერები. ის საღმას უსვამს  $Q$  რაოდენობის კითხვას, დანომრილს 0-დან  $(Q - 1)$ -მდე.  $k$ -ურ კითხვაში ( $0 \leq k < Q$ ) იასმინი უთითებს მოზაიკის ქვემართკუთხედს მისი:

- ზედა  $T[k]$  და ქვედა  $B[k]$  სტრიქონებით ( $0 \leq T[k] \leq B[k] < N$ );
- მარცხენა  $L[k]$  და მარჯვენა  $R[k]$  სვეტებით ( $0 \leq L[k] \leq R[k] < N$ ).

კითხვაზე პასუხი ქვემართკუთხედში შავი ფილების რაოდენობაა. ანუ, საღმამ უნდა გაიგოს რამდენი ისეთი  $(i, j)$  ფილა არსებობს, რომ  $T[k] \leq i \leq B[k]$ ,  $L[k] \leq j \leq R[k]$  და  $(i, j)$  ფილის ფერი არის შავი.

დანერეთ პროგრამა, რომელიც პასუხობს იასმინის კითხვებს.

# იმპლემენტაციის დეტალები

თქვენ უნდა მოახდინოთ შემდეგი პროცედურის იმპლემენტაცია.

```
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)
```

- $X, Y$ :  $N$  სიგრძის მასივები, რომლებიც აღწერენ ფილების ფერებს ზედა სტრიქონსა და მარცხენა სვეტში შესაბამისად;
- $T, B, L, R$ :  $Q$  სიგრძის მასივები, რომლებიც აღწერენ იასმინის კითხვებს;
- პროცედურამ უნდა დააბრუნოს  $Q$  სიგრძის  $C$  მასივი, სადაც  $C[k]$  არის პასუხი  $k$ -ურ კითხვაზე ( $0 \leq k < Q$ );
- ეს პროცედურა გამოძახებული იქნება ერთხელ ყოველი ტესტისთვის.

## შეზღუდვები

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 200\,000$
- $X[i] \in \{0, 1\}$  და  $Y[i] \in \{0, 1\}$  ყოველი  $i$ -სათვის, სადაც  $0 \leq i < N$
- $X[0] = Y[0]$
- $0 \leq T[k] \leq B[k] < N$  და  $0 \leq L[k] \leq R[k] < N$  ყოველი  $k$ -სათვის, სადაც  $0 \leq k < Q$

## ქვეამოცანები

ქვეამოცანა	ქულა	დამატებითი შეზღუდვები
1	5	$N \leq 2; Q \leq 10$
2	7	$N \leq 200; Q \leq 200$
3	7	$T[k] = B[k] = 0$ (ყოველი $k$ -სათვის, სადაც $0 \leq k < Q$ )
4	10	$N \leq 5000$
5	8	$X[i] = Y[i] = 0$ (ყოველი $i$ -სათვის, სადაც $0 \leq i < N$ )
6	22	$T[k] = B[k]$ და $L[k] = R[k]$ (ყოველი $k$ -სათვის, სადაც $0 \leq k < Q$ )
7	19	$T[k] = B[k]$ (ყოველი $k$ -სათვის, სადაც $0 \leq k < Q$ )
8	22	დამატებითი შეზღუდვების გარეშე.

## მაგალითი

განვიხილოთ შემდეგი გამოძახება.

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

ეს მაგალითი ნაჩვენებია სურათებზე ქვემოთ. მარცხენა სურათზე ნაჩვენებია მოზაიკის ფილების ფერები. შუა და მარჯვენა სურათებზე ნაჩვენებია ქვემართკუთხედები, რომლებიც ფიგურირებს იასმინის პირველ და მეორე კითხვებში შესაბამისად.

	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

კითხვებზე პასუხები (ანუ, დაჩრდილულ მართკუთხედებში ერთების რაოდენობა) არის 7 და 3 შესაბამისად. მაშასადამე, პროცედურამ უნდა დააბრუნოს [7, 3].

## სანიმუშო გრაფერი

შეტანის ფორმატი:

```
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]
```

გამოტანის ფორმატი:

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

აქ  $S$  არის `mosaic`-ის მიერ დაბრუნებული  $C$  მასივის სიგრძე.