

A. Mengecat Dinding

Time limit	1500 ms
Memory limit	512 MB

Deskripsi

Sudah lama sejak terakhir kali Pak Dengklek mengecat dinding rumahnya, sehingga Pak Dengklek ingin mengecat ulang dindingnya. Dinding rumah Pak Dengklek terdiri atas N segmen, yang dinomori dari 0 hingga $N - 1$. Untuk persoalan ini, asumsikan kita memiliki K warna berbeda, yang direpresentasikan oleh angka dari 0 hingga $K - 1$ (contohnya, merah dinyatakan oleh 0, biru dinyatakan oleh 1, dan seterusnya). Pak Dengklek ingin mewarnai dinding ke- i dengan warna $C[i]$.

Untuk mewarnai dindingnya, Pak dengklek menyewa sebuah perusahaan konstruksi dengan M kontraktor, yang dinomori dari 0 hingga $M - 1$. Sayangnya, setiap kontraktor hanya ingin mewarnai dengan warna yang mereka suka. Secara spesifik, kontraktor ke- j hanya menyukai $A[j]$ warna dan hanya ingin mewarnai segmen yang harus diberi warna dengan salah satu di antara warna: warna $B[j][0]$, warna $B[j][1]$, ..., atau warna $B[j][A[j] - 1]$.

Pak Dengklek dapat memberikan beberapa instruksi kepada perusahaan konstruksi. Dalam sebuah instruksi, Pak Dengklek akan memberikan dua parameter x dan y , dimana $0 \leq x < M$ dan $0 \leq y \leq N - M$. Kemudian, perusahaan konstruksi akan memerintah kontraktor ke- $((x + l) \bmod M)$ untuk mewarnai segmen ke- $(y + l)$, dimana $0 \leq l < M$. Apabila terdapat sebuah nilai l sehingga kontraktor ke- $((x + l) \bmod M)$ tidak menyukai warna $C[y + l]$, maka instruksi dianggap tidak valid.

Pak Dengklek harus membayar untuk setiap instruksi yang ia berikan, sehingga Pak Dengklek ingin mengetahui banyaknya instruksi minimum agar dapat mewarnai seluruh segmen dengan warna yang diinginkan atau periksa apabila hal tersebut tidak mungkin untuk dilakukan. Sebuah segmen dapat diwarnai beberapa kali, namun harus selalu diwarnai dengan warna yang diinginkan.

Tugas

Anda diharuskan mengimplementasikan fungsi `minimumInstructions`:

- `minimumInstructions(N, M, K, C, A, B)` - Fungsi ini akan dipanggil hanya sekali oleh *grader*.
 - N : Sebuah bilangan bulat yang menyatakan banyak segmen.
 - M : Sebuah bilangan bulat yang menyatakan banyak kontraktor.
 - K : Sebuah bilangan bulat yang menyatakan banyak warna.
 - C : Sebuah *array* berisi N bilangan bulat menyatakan warna yang diinginkan untuk sebuah segmen.
 - A : Sebuah *array* berisi M bilangan bulat menyatakan banyak warna yang disukai oleh kontraktor.
 - B : Sebuah *array* berukuran M berisi *array* bilangan bulat, menyatakan warna yang disukai oleh kontraktor.
 - Fungsi ini harus mengembalikan sebuah bilangan bulat menyatakan banyak instruksi minimum yang harus diberikan Pak Dengklek untuk mengecat seluruh segmen berdasarkan warna yang diinginkan, atau -1 jika tidak mungkin untuk melakukan hal tersebut.

Contoh

Pada contoh pertama, $N = 8$, $M = 3$, $K = 5$, $C = [3, 3, 1, 3, 4, 4, 2, 2]$, $A = [3, 2, 2]$, $B = [[0, 1, 2], [2, 3], [3, 4]]$. Pak Dengklek dapat memberikan instruksi berikut ini:

- $x = 1$, $y = 0$. Ini merupakan instruksi yang valid karena kontraktor pertama dapat mewarnai segmen ke-0, kontraktor kedua dapat mewarnai segmen pertama, dan kontraktor ke-0 dapat mewarnai segmen kedua.

2. $x = 0, y = 2$. Ini merupakan instruksi yang valid karena kontraktor ke-0 dapat mewarnai segmen kedua, kontraktor pertama dapat mewarnai segmen ketiga, dan kontraktor kedua dapat mewarnai segmen keempat.
3. $x = 2, y = 5$. Ini merupakan instruksi yang valid karena kontraktor kedua dapat mewarnai segmen kelima, kontraktor ke-0 dapat mewarnai segmen keenam, dan kontraktor pertama dapat mewarnai segmen ketujuh.

Jelas bahwa Pak Dengklek tidak dapat memberikan kurang dari 3 instruksi untuk mewarnai semua segmen berdasarkan warna yang diharapkan, sehingga `minimumInstructions(8, 3, 5, [3, 3, 1, 3, 4, 4, 2, 2], [3, 2, 2], [[0, 1, 2], [2, 3], [3, 4]])` harus mengembalikan 3.

Pada contoh kedua, $N = 5, M = 4, K = 4, C = [1, 0, 1, 2, 2], A = [2, 1, 1, 1], B = [[0, 1], [1], [2], [3]]$.

Dikarenakan kontraktor ketiga hanya menyukai warna 3, sedangkan tidak ada segmen yang harus diwarnai dengan warna 3, maka mustahil bagi Pak Dengklek untuk memberikan instruksi yang valid. Oleh karena itu, `minimumInstructions(5, 4, 4, [1, 0, 1, 2, 2], [2, 1, 1, 1], [[0, 1], [1], [2], [3]])` harus mengembalikan -1 .

Batasan

Untuk $0 \leq k < K$, misalkan $f(k)$ merupakan banyaknya bilangan j sehingga kontraktor ke- j menyukai warna k . Sebagai contoh, apabila $f(1) = 2$, maka terdapat dua kontraktor yang menyukai warna 1.

- $1 \leq N \leq 100\,000$.
- $1 \leq M \leq \min(N, 50\,000)$.
- $1 \leq K \leq 100\,000$.
- $0 \leq C[i] < K$.
- $1 \leq A[j] \leq K$.
- $0 \leq B[j][0] < B[j][1] < \dots < B[j][A[j] - 1] < K$.
- Hasil jumlah $f(k)^2 \leq 400\,000$.

Subsoal 1 (12 poin)

- $f(k) \leq 1$.

Subsoal 2 (15 poin)

- $N \leq 500$.
- $M \leq \min(N, 200)$.
- Hasil jumlah $f(k)^2 \leq 1\,000$.

Subsoal 3 (13 poin)

- $N \leq 500$.
- $M \leq \min(N, 200)$.

Subsoal 4 (23 poin)

- $N \leq 20\,000$.
- $M \leq \min(N, 2\,000)$.

Subsoal 5 (37 poin)

- Tidak ada batasan tambahan.

Contoh Grader

Contoh *grader* yang diberikan akan membaca masukan dengan format sebagai berikut:

N M K

C[0] C[1] ... C[N-1]

A[0] B[0][0] B[0][1] ... B[0][A[0]-1]

A[1] B[1][0] B[1][1] ... B[1][A[1]-1]

▪

▪

▪

A[M-1] B[M-1][0] B[M-1][1] ... B[M-1][A[M-1]-1]

Contoh *grader* akan mencetak nilai kembalian dari fungsi `minimumInstructions`.