

Contador de Sub-Secuencias Únicas (Aplicando Hashing y Ventana Deslizante)

Descripción del Problema

Se dispone de una secuencia de N números enteros y un valor entero K. El objetivo consiste en determinar cuántas sub-secuencias contiguas de longitud exactamente K presentan únicamente elementos distintos, es decir, sin repeticiones dentro de dicha ventana.

Detalles y Restricciones

Entrada:

- La primera línea contiene los valores N y K, donde $1 \leq K \leq N \leq 100000$.
- La segunda línea contiene N enteros, a_1, a_2, \dots, a_n , cuyos valores se encuentran entre -1000000000 y 1000000000 .

Salida:

- Se debe imprimir un único entero: la cantidad total de sub-secuencias contiguas de longitud K que contienen exclusivamente elementos no repetidos.

Ejemplo

Input	Output
7 3 1 2 3 2 4 5 4	3

Explicación del Ejemplo

La secuencia analizada es:

[1, 2, 3, 2, 4, 5, 4], con una longitud deseada K = 3.

Se evalúan todas las sub-secuencias contiguas de tamaño 3:

Posiciones Sub-secuencia Elementos Repetidos ¿Es Única?

0 a 2	[1, 2, 3]	Ninguno	Sí
1 a 3	[2, 3, 2]	2 se repite	No
2 a 4	[3, 2, 4]	Ninguno	Sí
3 a 5	[2, 4, 5]	Ninguno	Sí
4 a 6	[4, 5, 4]	4 se repite	No

En total, existen 3 sub-secuencias válidas que cumplen el criterio de unicidad.

CÓDIGO

```
#include <iostream>
using namespace std;

int main() {
    int N, K;
    cin >> N >> K;

    // Arreglo dinámico para los números (sin [])
    int* arr = new int[N];

    // Lectura usando punteros
    for (int i = 0; i < N; i++) {
        cin >> *(arr + i);
    }

    // Mapa hash simple usando arreglo dinámico (para frecuencia)
    int* freq = new int[N];
    for (int i = 0; i < N; i++) {
        *(freq + i) = 0;
    }

    int conteoUnicas = 0;
    int repetidos = 0;

    auto agregar = [&](int valor, int index) {
        int* f = freq + index;
        if (*f == 1) repetidos++;
        (*f)++;
    };

    auto quitar = [&](int valor, int index) {
```

```
int* f = freq + index;

if (*f == 2) repetidos--;
(*f)--;
};

// Primera ventana

for (int i = 0; i < K; i++) {
    agregar(*(arr + i), i);
}

if (repetidos == 0) conteoUnicas++;

// Recorrer el resto con ventana deslizante

for (int i = K; i < N; i++) {
    quitar(*(arr + (i - K)), i - K);
    agregar(*(arr + i), i);

    if (repetidos == 0) conteoUnicas++;
}

cout << conteoUnicas << endl;

delete[] arr;
delete[] freq;

return 0;
}
```