URL VIDEOS

Explicación enunciado del problema:https://youtu.be/qxqufXWqemc
Explicación código: https://youtu.be/FJIstRDc4Ws
Análisis de complejidad: https://youtu.be/XLVDRXECkcw

ENUNCIADO DEL PROBLEMA

Dado un arreglo a de longitud n y un entero k, hallar el número de índices $1 \leq i \leq n-k$ tales que la submatriz [ai,...,ai+k] con longitud k+1 (no con longitud k) tiene la siguiente propiedad:

Si se multiplica el primer elemento por 2^0, el segundo elemento por 2^1, ..., y el (k+1) elemento por 2^k, entonces este sub arreglo está ordenado en orden estrictamente creciente. Más formalmente, contar el número de índices $1 \leq i \leq n-k$ tales que

$2^0 \cdot ai < 2^1 \cdot ai+1 < 2^2 \cdot ai+2 < \cdots < 2^k \cdot ai+k$.

Input

La primera línea contiene dos enteros, n y k (3 <= n <= 2*10^5, 1<= k < n), la longitud del arreglo y el número de desigualdades.

La segunda línea contiene n enteros a1, a2,...,an (1 <= ai <= 10^9), los cuales representan los elementos del arreglo

Output

Para cada caso de prueba, imprimir un solo entero, el numero de indices que satisfagan la condición del enunciado

URL DE GITLAB
https://gitlab.com/ejercicios-proyecto-ada/ejercicio-4-2sort

MÉTODO CÁNDIDO

```java
public int metodoCandido(int n, int k, int arr[]) {
int valido, respuesta = 0;
for(int i = 0; i⩽n-k-1; i++) {
        valido = 0;
        for(int j = i; j<i+k; j++) {
            if (arr[j] < 2*arr[j+1]) {
                valido++;
            }
        }
        if (valido == k) {
            respuesta++;
        }
    }
    return respuesta;
}
```

Eficacia	El método es eficaz, puesto a que tiene una solución para todas las entradas válidas ingresadas.

	Instancia 1: LONGITUD DEL ARREGLO: 7 K: 2 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 22, 12 RESULTADO MÉTODO CÁNDIDO: 3
	Instancia 2: LONGITUD DEL ARREGLO: 20 K: 3 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 2, 12, 54, 64, 12, 14, 45, 65, 86, 3, 14, 65, 87, 5, 10 RESULTADO MÉTODO CÁNDIDO: 5
	Instancia 3: LONGITUD DEL ARREGLO: 200000 K: 100 ARREGLO DE NÚMEROS: …. RESULTADO MÉTODO CÁNDIDO: 199900
Eficiencia	El método no es el más eficiente ya que analiza los elementos múltiples veces, haciendo que avance muy

	lentamente en la lista de números y requiere de esto para conseguir el resultado correcto; esto se ve reflejado en la instancia N°3 donde hay una mayor cantidad de elementos y esta falencia toma mayor peso.
	Instancia 1: LONGITUD DEL ARREGLO: 7 K: 2 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 22, 12 RESULTADO MÉTODO CÁNDIDO: 3 **Tiempo del proceso: 58 microsegundos**
	Instancia 2: LONGITUD DEL ARREGLO: 20 K: 3 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 2, 12, 54, 64, 12, 14, 45, 65, 86, 3, 14, 65, 87, 5, 10 RESULTADO MÉTODO CÁNDIDO: 5 **Tiempo del proceso: 80 microsegundos**
	Instancia 3: LONGITUD DEL ARREGLO: 200000 K: 100

	ARREGLO DE NÚMEROS: …. RESULTADO MÉTODO CÁNDIDO: 199900 **Tiempo del proceso: 17003 microsegundos**
Correctitud	El algoritmo es correcto, ya que da un resultado esperado para todas las entradas posibles
	Instancia 1: LONGITUD DEL ARREGLO: 7 K: 2 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 22, 12 RESPUESTA ESPERADA: 3 RESULTADO MÉTODO CÁNDIDO: 3
	Instancia 2: LONGITUD DEL ARREGLO: 20 K: 3 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 2, 12, 54, 64, 12, 14, 45, 65, 86, 3, 14, 65, 87, 5, 10 RESPUESTA ESPERADA: 5 RESULTADO MÉTODO CÁNDIDO: 5
	Instancia 3:

	LONGITUD DEL ARREGLO: 200000 K: 100 ARREGLO DE NÚMEROS: …. RESULTADO ESPERADO: 199900 RESULTADO MÉTODO CÁNDIDO: 199900
Completitud	El método es completo, ya que la lógica usada para hallar la cantidad de subarreglos que cumplen la condición, aunque es lento asegura siempre un resultado correcto, si es que este existe
	Instancia 1: LONGITUD DEL ARREGLO: 7 K: 2 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 22, 12 RESULTADO ESPERADO: 3 RESULTADO MÉTODO CÁNDIDO: 3
	LONGITUD DEL ARREGLO: 20 K: 3 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 2, 12, 54, 64, 12, 14, 45, 65, 86, 3, 14, 65, 87, 5, 10 RESULTADO ESPERADO: 5

	RESULTADO MÉTODO CÁNDIDO: 5
	Instancia 3: LONGITUD DEL ARREGLO: 200000 K: 100 ARREGLO DE NÚMEROS: …. RESULTADO ESPERADO: 199900 RESULTADO MÉTODO CÁNDIDO: 199900

MÉTODO 2	
`` public int metodoOptimo(int n, int k, int arr[]) { int valido = 0; int respuesta = 0; for(int i = 0; i<n-1; i++) { if(arr[i] < 2*arr[i+1]) { valido++; } else { valido = 0; } if (valido ≥ k) { respuesta++; } } return respuesta; } ```	
Eficacia	El método es eficaz, puesto a que tiene una solución para todas las entradas válidas ingresadas
	Instancia 1: LONGITUD DEL ARREGLO: 7

	K: 2 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 22, 12 RESULTADO MÉTODO ÓPTIMO: 3
	Instancia 2: LONGITUD DEL ARREGLO: 20 K: 3 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 2, 12, 54, 64, 12, 14, 45, 65, 86, 3, 14, 65, 87, 5, 10 RESULTADO MÉTODO ÓPTIMO: 5
	Instancia 3: LONGITUD DEL ARREGLO: 200000 K: 100 ARREGLO DE NÚMEROS: …. RESULTADO MÉTODO ÓPTIMO: 199900
Eficiencia	El método es eficiente ya que solo necesita de un recorrido para llegar a un resultado correcto, evitando las múltiples validaciones de los mismo valores
	Instancia 1: LONGITUD DEL ARREGLO: 7 K: 2

	ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 22, 12 RESULTADO MÉTODO ÓPTIMO: 3 **Tiempo del proceso: 27 microsegundo**
	Instancia 2: LONGITUD DEL ARREGLO: 20 K: 3 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 2, 12, 54, 64, 12, 14, 45, 65, 86, 3, 14, 65, 87, 5, 10 RESULTADO MÉTODO ÓPTIMO: 5 **Tiempo del proceso: 24 microsegundos**
	Instancia 3: LONGITUD DEL ARREGLO: 200000 K: 100 ARREGLO DE NÚMEROS: …. RESULTADO MÉTODO ÓPTIMO: 199900 **Tiempo del proceso: 4945 microsegundos**
Correctitud	El algoritmo es correcto, ya que da un resultado esperado para todas las entradas posibles
	Instancia 1:

	LONGITUD DEL ARREGLO: 7 K: 2 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 22, 12 RESULTADO ESPERADO: 3 RESULTADO MÉTODO ÓPTIMO: 3
	Instancia 2: LONGITUD DEL ARREGLO: 20 K: 3 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 2, 12, 54, 64, 12, 14, 45, 65, 86, 3, 14, 65, 87, 5, 10 RESULTADO ESPERADO: 5 RESULTADO MÉTODO ÓPTIMO: 5
	Instancia 3: LONGITUD DEL ARREGLO: 200000 K: 100 ARREGLO DE NÚMEROS: …. RESULTADO ESPERADO: 199900 RESULTADO MÉTODO ÓPTIMO: 199900
Completitud	El método es completo, ya que la lógica usada para hallar la cantidad de subarreglos que cumplen la condición, aunque es lento asegura siempre un resultado correcto, si es que este existe

	Instancia 1: LONGITUD DEL ARREGLO: 7 K: 2 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 22, 12 RESULTADO ESPERADO: 3 RESULTADO MÉTODO ÓPTIMO: 3
	Instancia 2: LONGITUD DEL ARREGLO: 20 K: 3 ARREGLO DE NÚMEROS: 22, 12, 16, 4, 3, 2, 12, 54, 64, 12, 14, 45, 65, 86, 3, 14, 65, 87, 5, 10 RESULTADO ESPERADO: 5 RESULTADO MÉTODO ÓPTIMO: 5
	Instancia 3: LONGITUD DEL ARREGLO: 200000 K: 100 ARREGLO DE NÚMEROS: …. RESULTADO ESPERADO: 199900 RESULTADO MÉTODO ÓPTIMO: 199900