

Ejercicio 1 — Posición de corte en un array rotado

Enunciado:

Dado un arreglo ordenado ascendentemente que ha sido rotado un número desconocido de veces (sin elementos duplicados), encuentra el índice del menor elemento (el punto de corte). Implementa una solución con complejidad $O(\log n)$.

Formato de entrada:

Primera línea: n ($1 \leq n \leq 2 \cdot 10^5$)

Segunda línea: n enteros que representan el arreglo rotado.

Formato de salida:

Un único entero: el índice (0-based) del elemento mínimo.

Ejemplo:

Entrada:

7

13 18 25 2 6 8 11

Salida:

3

Ejercicio 2 — Primera posición con valor $\geq X$

Enunciado:

Dado un arreglo ordenado no decreciente de enteros (puede tener repeticiones), y un entero X , encuentra el índice de la primera posición cuya valor sea mayor o igual a X . Si no existe tal posición, devuelve -1. Debe resolverse en $O(\log n)$.

Formato de entrada:

Primera línea: n ($1 \leq n \leq 2 \cdot 10^5$) y X

Segunda línea: n enteros ordenados no decrecientemente.

Formato de salida:

Un entero: índice 0-based de la primera aparición con valor $\geq X$, o -1 si no existe.

Ejemplo:

Entrada:

6 15

3 7 15 15 20 24

Salida:

2

Ejercicio 3 — Separar Wi-Fi: maximizar la mínima distancia

Enunciado:

Tienes M puntos disponibles (enteros) donde puedes instalar repetidores Wi-Fi en una línea (posiciones distintas), y debes instalar K repetidores en algunas de esas posiciones.

Encuentra la máxima distancia D tal que la distancia mínima entre cualesquiera dos repetidores sea al menos D. Diseña una solución con $O(M \log R)$ donde R es el rango de búsqueda.

Formato de entrada:

Primera línea: M ($1 \leq M \leq 2 \cdot 10^5$) y K ($1 \leq K \leq M$)

Segunda línea: M enteros distintos — las posiciones disponibles (no necesariamente ordenadas).

Formato de salida:

Un entero: la mayor D posible (entero) que se puede lograr.

Ejemplo:

Entrada:

5 3

1 2 8 4 9

Salida:

3

Ejercicio 4 — Mínima velocidad para completar tareas

Enunciado:

Un trabajador debe completar N tareas; la i-ésima tarea requiere t_i unidades de trabajo.

Con una velocidad V (tareas por unidad de tiempo), el tiempo para completar la tarea i es $\lceil t_i / V \rceil$. Encuentra la mínima velocidad entera $V \geq 1$ tal que el total de tiempo para completar todas las tareas no excede T dada. Resuelve con $O(N \log U)$, donde U es el límite superior buscado.

Formato de entrada:

Primera línea: N ($1 \leq N \leq 2 \cdot 10^5$) y T ($1 \leq T \leq 10^{18}$)

Segunda línea: N enteros t_i ($1 \leq t_i \leq 10^9$)

Formato de salida:

Un entero: la mínima V que cumple la condición, o -1 si no es posible (aunque con V grande siempre es posible en este modelo).

Ejemplo:

Entrada:

3 7

10 10 10

Salida:

5

Ejercicio 5 — Ejercicio de Hashing (Nivel Básico)

Enunciado:

Implementa una pequeña tabla hash para guardar pares (clave, valor), donde la clave es un número entero y el valor también.

La tabla debe permitir:

1. Insertar una pareja (clave, valor).
2. Buscar una clave y devolver su valor.
3. Usar la función hash: $h(k) = k \% 100$.
4. Si hay colisión, simplemente reemplazar el valor.

Entrada:

- Primera línea: un número N (cantidad de operaciones).
- Luego N líneas con dos tipos de operación:
 - 1 clave valor → insertar
 - 2 clave → buscar

Salida:

Para cada operación de tipo 2, imprimir el valor correspondiente o -1 si no existe.

Ejemplo:

Entrada:

5

1 10 555

1 110 999

2 10

2 110

2 210

Salida:

999

999

999