

Ejercicios de Divide y Vencerás

1. Contar Ceros en un Arreglo Binario

Dado un arreglo compuesto solo por 0 s y 1 s, donde todos los 1 s aparecen antes que los 0 s, encontrar el número total de ceros.

Ejemplos:

Entrada: {1, 1, 1, 0, 0, 0, 0}

Salida: 4

Entrada: {1, 1, 1, 1, 1}

Salida: 0

Tip:

Utiliza búsqueda binaria para encontrar el primer 0 . Luego, el número de ceros será igual a $n - \text{índice}$.

2. Elemento Único en un Arreglo Ordenado

Dado un arreglo ordenado en el cual todos los elementos aparecen dos veces, excepto uno, encontrar el elemento único. Complejidad requerida: $O(\log n)$.

Ejemplo:

Entrada: {1, 1, 3, 3, 4, 5, 5, 7, 7, 8, 8}

Salida: 4

Tip:

Aplica búsqueda binaria comparando pares en posiciones pares/impares. El elemento único estará del lado donde la paridad se rompe.

3. Mínimo en un Arreglo Rotado

Dado un arreglo rotado en un punto desconocido (sin elementos duplicados), encontrar el elemento mínimo.

Ejemplo:

Entrada: {5, 6, 1, 2, 3, 4}

Salida: 1

Tip:

Usa búsqueda binaria para detectar el punto de rotación, comparando el elemento medio con los extremos del subarreglo.

4. Máximo en un Arreglo Primero Creciente y Luego Decreciente

Dado un arreglo que primero **crece** y luego **decrece**, encontrar el **valor máximo**.

Ejemplo:

```
Entrada: {8, 10, 20, 80, 100, 200, 400, 500, 3, 2, 1}
Salida: 500
```

Tip:

Aplica una búsqueda tipo binaria para encontrar el pico (`peak`) comparando el elemento medio con sus vecinos.

5. Número de Inversiones en un Arreglo

Dado un arreglo de enteros, encontrar el **número total de inversiones**.

Una **inversión** es un par `(A[i], A[j])` tal que `i < j` y `A[i] > A[j]`.

Ejemplo:

```
Entrada: {2, 4, 1, 3, 5}
Salida: 3
```

Inversiones: (2,1), (4,1), (4,3)

Tip:

Usa un **merge sort modificado** que cuente inversiones en cada paso de mezcla. Cada vez que tomas un elemento del subarreglo derecho antes que uno del izquierdo, cuentas una inversión.

6. Suma Máxima de Subarreglo Contiguo

Dado un arreglo que contiene enteros positivos y negativos, encontrar la **máxima suma de un subarreglo contiguo**.

Ejemplo:

```
Entrada: {-2, -5, 6, -2, -3, 1, 5, -6}
Salida: 7
```

Subarreglo con suma máxima: `{6, -2, -3, 1, 5}`

Tip:

Divide el arreglo en dos mitades:

- Calcula la suma máxima en la mitad izquierda
- Calcula la suma máxima en la mitad derecha
- Calcula la suma máxima **cruzando el centro**

Devuelve el máximo de los tres. Complejidad: `O(n log n)`

7. Ordenación Rápida (QuickSort)

Implementa el algoritmo de **ordenación rápida** (quicksort) para ordenar un arreglo de enteros en orden ascendente.

Ejemplo:

Entrada: {10, 7, 8, 9, 1, 5}

Salida: {1, 5, 7, 8, 9, 10}

Tip:

Selecciona un pivote, particiona el arreglo en elementos menores y mayores al pivote, y aplica recursivamente el mismo proceso a cada subarreglo.