



Instituto Politécnico Nacional



Escuela Superior de Cómputo

Análisis y Diseño de Algoritmos

Mercado Tarango Luis Leonardo

3CV6

Reporte de Resultados del Programa

Descripción del Problema

Dado un arreglo ordenado de forma ascendente, se requiere remover los duplicados en el sitio, de manera que cada elemento único aparezca solo una vez. El orden relativo de los elementos debe mantenerse, y se debe retornar el número de elementos únicos.

Implementación en C

El programa se implementó en C y realiza las siguientes acciones:

1. Define la función `removeDuplicates` que toma un puntero al arreglo `nums` y su tamaño `numsSize`.
2. Itera a través del arreglo y mueve los elementos únicos hacia el frente del arreglo.
3. Retorna el número de elementos únicos.
4. En la función `main`, se llama a `removeDuplicates` y se imprime el número de elementos únicos y el arreglo modificado.

Resultados

Ejemplo 1

- Entrada: `nums = [1, 1, 2]`
- Salida Esperada: `2, nums = [1, 2, _]`
- Salida del Programa:
 - Número de elementos únicos: 2
 - Arreglo después de remover duplicados: `[1, 2]`

Ejemplo 2

- Entrada: `nums = [0, 0, 1, 1, 1, 2, 2, 3, 3, 4]`
- Salida Esperada: `5, nums = [0, 1, 2, 3, 4, _, _, _, _, _]`
- Salida del Programa:
 - Número de elementos únicos: 5
 - Arreglo después de remover duplicados: `[0, 1, 2, 3, 4]`

Complejidad del Algoritmo

- Complejidad Temporal: $O(n)$, donde n es el tamaño del arreglo. Esto se debe a que recorremos el arreglo una sola vez.
- Complejidad Espacial: $O(1)$, ya que no utilizamos espacio adicional significativo aparte de unas pocas variables auxiliares.

Conclusión

El programa cumple con los requisitos especificados, removiendo duplicados en el sitio y retornando el número de elementos únicos. La implementación es eficiente tanto en términos de tiempo como de espacio.