Universidad Francisco de Paula Santander

Vigilada Mineducación

TALLER PROBLEMAS BÚSQUEDA Y ORDENAMIENTO "REMOVE DUPLICATES FROM SORTED ARRAY"

YOFER NICOLAS MORALES CASTAÑEDA

1152154

ANÁLISIS DE ALGORITMOS

MILTON JESUS VERA CONTRERAS

INGENIERÍA DE SISTEMAS

FRANCISCO DE PAULA SANTANDER

CÚCUTA, N-SANTANDER

2023

INTRODUCCIÓN

En este informe se presentará la solución al problema "Remove Duplicates from Sorted Array" de la plataforma LeetCode, así como una aplicación que genera casos de prueba y una aplicación completa con main y lectura de datos que cubre todas las posibilidades de casos del problema, de manera equilibrada y que se puede ejecutar y probar en cualquier IDE independiente de la plataforma. Se realizará un video explicando en detalle los tres numerales anteriores.

TALLER PROBLEMAS BÚSQUEDA Y ORDENAMIENTO

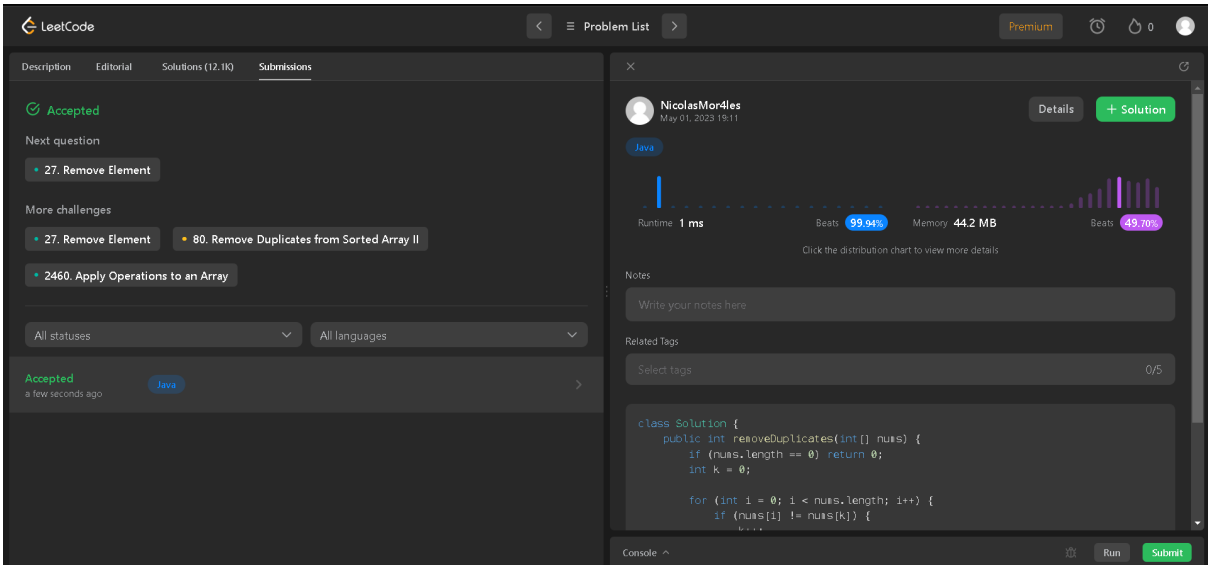
1) Resolver el problema en la plataforma: https://leetcode.com/
El ejercicio que se me asignó fue **“Remove Duplicates from Sorted Array”**
https://leetcode.com/problems/remove-duplicates-from-sorted-array/
Adjunto evidencia del problema resuelto y aceptado en la plataforma LeetCode:

Código:

```java
class Solution {
    public int removeDuplicates(int[] nums) {
        if (nums.length == 0) return 0;
        int k = 0;

        for (int i = 0; i < nums.length; i++) {
            if (nums[i] != nums[k]) {
                k++;
                nums[k] = nums[i];
            }
        }
        return k+1;
    }
}
```

Evidencia:



2) Desarrollar una aplicación que genere al menos 100 casos de prueba para el problema. Los casos deben cubrir todas las posibilidades de casos del problema, de manera equilibrada.

- **Caso 1:** Matriz con elementos positivos repetidos
- **Caso 2:** Matriz con elementos negativos repetidos
- **Caso 3:** Matriz con elementos positivos no repetidos
- **Caso 4:** Matriz con elementos negativos no repetidos

Código:

```
import java.util.Arrays;
import java.util.Random;

public class CasosDePrueba {
    public static void main(String[] args) {
        Random aleatorio = new Random();
        for (int i = 1; i <= 100; i++) {
            int longitud = aleatorio.nextInt(20) + 1;
            if (aleatorio.nextInt(10) == 0) {
                longitud = 100;
            }
            int[] numeros = new int[longitud];
            for (int j = 0; j < longitud; j++) {
                if (j > 0 && aleatorio.nextInt(4) == 0) {
                    numeros[j] = numeros[j-1];
                } else {
                    numeros[j] = aleatorio.nextInt(201) - 100;
                }
            }
            Arrays.sort(numeros);
            for (int j = 0; j < longitud; j++) {
                System.out.print(numeros[j] + ",");
            }
            System.out.println();
        }
    }
}
```

3) Desarrollar una aplicación completa, con main y lectura de datos, de manera que se pruebe la solución independientemente de la plataforma https://leetcode.com/.
Esta solución debe ser consistente con la solución realizada en el numeral 1.

Código:

```
import java.util.Arrays;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.println("Entrada:");
        String input = scanner.nextLine().replaceAll("\\[|\\]","");

        int[] nums =
Arrays.stream(input.split(",")).mapToInt(Integer::parseInt).toArray();
        int k = eliminarDuplicados(nums);
        System.out.print("Salida: " + k + ", nums = [");
        Arrays.stream(nums, 0, k).forEach(e -> System.out.print(e +
","));
        for (int i = k; i < nums.length; i++) {
            System.out.print("_,");
        }
        System.out.print("]");
    }

    public static int eliminarDuplicados(int[] nums) {
        if (nums.length == 0) return 0;
        int k = 0;

        for (int i = 0; i < nums.length; i++) {
            if (nums[i] != nums[k]) {
                k++;
                nums[k] = nums[i];
            }
        }
        return k+1;
    }
}
```

4) Elaborar un video explicando en detalle los tres numerales anteriores. Preferiblemente subirlo a Youtube.

- Link al video: https://youtu.be/XoKYv1QcArA

5) Publicar en github todo lo anterior y escribir un informe en PDF que se sube a UVIRTUAL con los link a todo lo anterior.

- Link GitHub: https://github.com/NicolasMorales54/Taller-Problemas-Busqueda-y-Ordenamiento---26.Remove-Duplicates-from-Sorted-Array

CONCLUSIÓN

El problema "Remove Duplicates from Sorted Array" de LeetCode se puede resolver utilizando dos punteros para eliminar los elementos duplicados de un array ordenado y devolver la longitud del nuevo array. Además, se presentó una aplicación que genera casos de prueba de manera equilibrada, lo que es esencial para probar la solución de manera completa. Finalmente, se desarrolló una aplicación completa con main y lectura de datos para probar la solución de manera independiente de la plataforma LeetCode.