
Estructuras de datos

Semestre 2024-1

Práctica 3

Profesor:

José Luis Vázquez Lázaro

Fecha de entrega Práctica: Martes 12 de septiembre
antes de las 12:00.

Fecha de entrega Reporte: Martes 12 de septiembre.

Ayudantes:

Erik Quintero Villeda

David Román Valencia Rodríguez

1. Problema

Dada una secuencia S no vacía y finita de n números enteros ordenados de manera creciente, eliminar los elementos repetidos de la secuencia.

Ejemplo:

Entrada: $[-1, 3, 3, 3, 3]$

Salida: $[-1, 3]$

2. Instrucciones:

- Léa cuidadosamente los requisitos de entrega de la práctica.

2.1. Práctica (Martes 12 de septiembre antes de las 12:00)

- Considerando el problema definido anteriormente, diseñe e implemente un algoritmo que resuelva el problema.
- La práctica deberá estar correctamente documentada con javadoc.
- Ejecuta tu algoritmo utilizando los siguientes arreglos y mide el tiempo que utiliza tu computadora para ejecutar el método en milisegundos (en caso de que tu computadora marque 0, mide el tiempo en nanosegundos y realiza la conversión). Les pasamos una carpeta con los arreglos, para facilitar un poco la práctica:
 1. $[-1, 3, 3, 3, 3]$
 2. $[2, 7, 7, 8, 8, 11, 11, 16, 16, 18]$
 3. $[2, 2, 3, 5, 5, 7, 9, 9, 10, 10, 12, 12, 14, 17, 17]$
 4. $[0, 0, 0, 0, 0, 1, 2, 3, 3, 3, 5, 5, 6, 7, 8, 8, 8, 8, 9, 10]$
 5. $[1, 2, 2, 2, 2, 3, 3, 3, 4, 5, 5, 6, 6, 7, 7, 8, 9, 9, 10, 10, 11, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 20, 21, 21, 21, 21, 21, 23, 23, 23, 23, 23, 24, 25]$
- El algoritmo debe estar implementado dentro de la clase 'Clase_seis'.
- La práctica deberá tener un archivo README con extensión txt, donde se especifique:
 - como compilar y ejecutar la práctica
 - los nombres de los integrantes del equipo.
- Se deberá entregar en un archivo comprimido zip, este debe contener todos los archivos necesarios para que su práctica funcione. El nombre del zip deberá ser: apellidoIntegrante1_apellidoIntegrante2_practicax.zip, donde x es el número de la práctica.

2.2. Reporte (Martes 12 de septiembre)

- Elabora un reporte breve en el cual expliques cada uno de los siguientes puntos:
 - Describe detalladamente el diseño de tu algoritmo como se vio en clase. Indicando claramente si tu algoritmo regresará la entrada modificada o si devolverá un arreglo con los elementos de la entrada sin repetición.
 - Da un invariante que permita demostrar que el algoritmo es correcto (no es necesario dar la demostración, pero el invariante debe estar bien redactado).
 - Calcula el tiempo de ejecución $T(n)$ como se vio en clase de teoría y realiza una gráfica de tamaño de la entrada vs $T(n)$ para $n = 5, 10, 15, 20, 25$ (no se aceptarán gráficas de barras, pastel, etc, tiene que ser una gráfica de R^2).
 - Utilizando los tiempos obtenidos en el laboratorio, realiza una gráfica de tamaño de la entrada vs tiempo en milisegundos (si tu computadora muestra 0 al medir el tiempo, mídelo en nano segundos y realiza una conversión a milisegundos).
 - Compara la gráficas del tiempo de ejecución y el tiempo en milisegundos y escribe una conclusión sobre esta comparación (notemos que estamos comparando resultados prácticos con resultados teóricos).

Este reporte se entregará el día jueves 31 de agosto antes de las 12:00 pm (puedes hacer el reporte en

3. Requisitos Generales:

- Se deberá entregar en la plataforma classroom (por favor limpien su almacenamiento si no pueden).
- La práctica deberá ser entregada en parejas.
- El reporte se puede realizar en latex, word, o docs, la entrega tiene que ser en formato PDF.
- Las gráficas deben de realizarse en algún software para graficar (geogebra, etc).