

ESTRUCTURAS DE DATOS Y ALGORITMOS II

PROYECTO #1 – COMPLEJIDAD COMPUTACIONAL EN LOS ALGORITMOS DE ORDENAMIENTO

OBJETIVO

Que el alumno observe la complejidad computacional de los algoritmos de ordenamiento para comparar su eficiencia de ejecución en grandes volúmenes de información

DESCRIPCIÓN GENERAL

Se deberá trabajar con los algoritmos de ordenamiento InsertionSort, SelectionSort Heapsort, Bubblesort, Quicksort, Mergesort, y un algoritmo más para observar el comportamiento de la cantidad de operaciones de estos algoritmos conforme aumenta el tamaño de la entrada

Los alumnos realizarán un programa para evaluar los algoritmos de ordenamiento y un reporte de los resultados obtenidos en dichas evaluaciones

ACTIVIDADES DEL PROYECTO

- ✓ Agrega en los algoritmos de ordenamiento, las instrucciones necesarias para contabilizar el número de operaciones del algoritmo (comparaciones, intercambios, inserciones) que sean considerables en el análisis de complejidad temporal
- ✓ Realiza pruebas con arreglos de diferente tamaño (llenar los valores con números aleatorios)
 - 50, 100, 500, 800, 1000, 2000, 5000, 10'000*
- ✓ Realiza al menos 5 ejecuciones en cada uno de los tamaños mencionados y muestra una tabla general con promedio del número de operaciones en cada uno
*En el trabajo de investigación se deberán mostrar evidencias de que los datos fueron obtenidos por el programa realizado
- ✓ Realiza gráficas (en el programa que tu prefieras) de la cantidad de operaciones que se realizan de acuerdo con el tamaño de la entrada y explica tus resultados. En estas graficas observarás la complejidad de cada algoritmo
- ✓ Deberás realizar una gráfica para comparar insertion, selection y bubble y otra gráfica para comparar quick, heap y merge
(eje x: tamaño del arreglo, eje y: número de operaciones)
- ✓ Deberás agregar capturas de pantalla de los aspectos más relevantes de las pruebas realizadas

ACTIVIDADES EXTRAS

- ✓ Investiga un algoritmo de ordenamiento de complejidad $n \log n$ diferente a los vistos en clase y uno de complejidad n^2 también diferente a los vistos en clase
- ✓ Agrega a esos algoritmos la contabilización de operaciones
- ✓ Incluye en tus gráficas estos algoritmos adicionales
- ✓ Realizar el análisis para la contabilización de operaciones de los algoritmos Counting y radix sort
- ✓ Realiza una tercer gráfica para comparar estos algoritmos.

FORMATO DE ENTREGA

Se deberán incluir los siguientes elementos (Generar una carpeta por cada uno):

- **Trabajo escrito** del reporte del proyecto:
 - Objetivo
 - Introducción (opcional)
 - Desarrollo
 - Explicación de las modificaciones realizadas a cada algoritmo, principales dificultades y forma en que se resolvieron
 - Implementación. Graficas de los algoritmos de ordenamiento, capturas de pantalla y explicación de los resultados obtenidos
 - Tabla de resultados del número de operaciones
 - Conclusiones individuales.

El documento escrito deberá ser realizado con el procesador de textos LaTeX, y se deberá entregar el archivo .tex y el archivo .pdf generado en él.

- **Código fuente** del programa realizado. Cada uno de los archivos de código deberá tener una cabecera de comentarios generales: Autor, fecha de última modificación y propósito del archivo
- **Documentación.** La documentación se refiere a los elementos necesarios para realizar el programa
 - Material de apoyo utilizado por el alumno integrado por imágenes, libros, textos o cualquier material obtenido en internet o físicamente (*obligatorio*)
 - Diagramas de flujo y/o UML (*opcional*)
 - Propuesta de diseño de archivos o clases (*opcional*)
 - Calendarización del proyecto (*opcional*)
 - Bitácoras de reunión de los integrantes (*opcional*)
 - Carpeta de javadoc (*opcional en caso de realizar el programa en java*)
- **Video Explicativo.** Se deberá realizar un video explicativo mostrando la ejecución del programa al menos una vez para cada uno de los algoritmos de ordenamiento; mostrando los resultados del número de operaciones obtenidas, así mismo se deberá explicar de manera general la estructura del código y las modificaciones realizadas a los algoritmos. El video deberá tener una duración máxima de 20 min y es obligatorio que participen todos los integrantes del equipo de manera interactiva

Las carpetas anteriores se deberán subir en alguna plataforma de almacenamiento de archivos (OneDrive, Google Drive, Dropbox). Solo un integrante del equipo realiza la entrega

- **E1 Aguilar Lopez Perez Proy1** No poner _ en el nombre del archivo

OBSERVACIONES

- El programa deberá ser realizado en alguno de los siguientes lenguajes: JAVA // C
- Es obligatorio realizar el programa siguiendo la propiedad de modularidad (dividir en varios archivos y/o en bibliotecas)
- La fecha de entrega para el programa será **el 20 de septiembre de 2025**
- Está estrictamente prohibido el plagio de programas y el uso indiscriminado de la i.a. generativa
- Se permite que parte de los archivos sean basados en bibliografía siempre y cuando ésta se incluya de manera correcta y no sean copias idénticas.