

PRACTICA # 1: ALGORITMOS DE ORDENAMIENTO PARTE 1

Objetivo general: El estudiante identificará la estructura de los algoritmos de ordenamiento por selección y por inserción.

Objetivo de la clase: El alumno revisará el uso de bibliotecas como capas de abstracción así como realización de pruebas de los algoritmos para diferentes instancias de colecciones para conocer la complejidad de estos

ACTIVIDADES PARA EL DESARROLLO DE LA PRÁCTICA**Ejercicio 1. Análisis Inicial**

El concepto de capas de abstracción permite al programador elaborar sus propias bibliotecas basado en la solución de problemas específicos para poder elaborar soluciones más complejas.

a) Verifica los archivos proporcionados, junto con sus respectivas implementaciones:

- ✓ utilerias.h
- ✓ utilerías.c
- ✓ ordenamientos.h
- ✓ ordenamientos.h

En el reporte, realiza comentarios de los aspectos que te parezcan relevantes y de su funcionamiento.

Ejercicio 2. Probando los ordenamientos

a) Crea un nuevo proyecto para lenguaje c (Puede ser un proyecto de Xcode, codeblocks, o Dev c++). Elabora un programa en el que, utilizando las bibliotecas proporcionadas, utilices los algoritmos que se encuentran en la biblioteca de ordenamientos

En dicho programa, mediante un menú de usuario se pueda elegir entre alguno de ellos. (insertion, selection)

- ✓ Se deberá crear un arreglo de 10 elementos y se llenará de manera aleatoria
- ✓ Se realizará el ordenamiento por el algoritmo seleccionado
- ✓ El programa deberá mostrar el arreglo original, las modificaciones que se hacen al arreglo en cada iteración y el arreglo final ordenado

Deberás incluir en el reporte capturas de pantalla de la ejecución del programa. Y el análisis de los puntos importantes a considerar en su elaboración

Ejercicio 3. Ahora en Java

- ✓ Abre el proyecto “Práctica 1” utilizando el editor NetBeans, **compila y ejecuta el programa** para verificar el funcionamiento.
- ✓ En caso de tener problemas de compatibilidad con el editor de netbeans, puedes crear un proyecto nuevo en tu editor, o bien, ejecutar los programas desde la consola. En cualquier caso debes explicar lo realizado y colocar las evidencias respectivas en el reporte
- ✓ Revisa las clases proporcionadas y realiza los comentarios de los aspectos que te llamen la atención o te causen dudas del código java (codificación, ejecución, clases, etc).

Ejercicio 4. Análisis de la Complejidad computacional

- ✓ Agrega en los algoritmos de ordenamiento, las instrucciones necesarias para contabilizar el número de operaciones del algoritmo (comparaciones intercambios inserciones o cualquier operación considerable para un análisis de complejidad)
- ✓ Realiza pruebas con arreglos de diferentes tamaños de entrada 50, 100,500, 1000 elementos (de preferencia que sean llenados aleatoriamente) y verifica el crecimiento del número de operaciones.
- ✓ 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

5.- Elabora las conclusiones de tu práctica, recuerda que lo más importante es indicar si se cumplieron los objetivos