

Algoritmos y Estructura de Datos

Hoja de Trabajo 3, Algoritmos de Sort

Objetivos:

- a. Comparación de la complejidad de tiempo de corrida de algoritmos de sort.
- b. Utilización de Profilers para medición de tiempos de corrida.
- c. Control de versiones del programa.
- d. Emplear JUnit para casos de prueba.

“Desarrolle un programa que genere al azar y guarde en un archivo, números enteros. Debe generar hasta 3000 números. Luego utilice su programa para leer los datos de ese archivo, guardarlos en un arreglo y ordenarlos, usando cada uno de los algoritmos de sort. Utilice el Profiler para medir el tiempo empleado en el ordenamiento para: 10 números hasta 3000 números. El propósito será generar una gráfica con el eje Y mostrando el tiempo y el eje X mostrando la cantidad de números ordenados. Usted puede seleccionar los intervalos que considere adecuados en el eje X.”

Gráfica de Excel:

https://docs.google.com/document/d/17L9BnRzxInfCBKN09yNsGHWaYGCXZA2Lhs_gjYLslqew/edit?usp=sharing

Evidencia de Desarrollo (GitHub):

<https://github.com/adirnnn/HDT3-Sort-Implementation.git>

Análisis:

- *Tipo de Profiler:* JProfiler.
- *Como fue utilizado:* JProfiler fue utilizado para medir los tiempos de cada algoritmo de Sort. Básicamente, se realizó:
 - Configuración de entorno.
 - La iniciación de JProfiler y selección del PATH.
 - Configuración de la sesión de perfilamiento.
 - Ejecución del código (específicamente la clase del sort específico) con JProfiler corriendo.
 - Análisis de resultados de tiempos.
 - Gráfica.