

OBSERVACIONES DE LA PRÁCTICA

Estudiante 1-Pablo Pedreros-202112491

Estudiante 2-Laura Guiza-201920926

Ambientes de pruebas

	Máquina 1	Máquina 2
Procesadores	Intel(R) Core(TM) i7-6700 CPU @3.40GHz 3408Mhz	
Memoria RAM (GB)	8,00 GB	
Sistema Operativo	Windows 10 Enterprise 64-bits	

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

Máquina 1

Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAYLIST)	Insertion Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
-small	768	265.63	15.63	15.63	15.63
5.00%	7572	26218.75	203.13	500.0	187.5
10.00%	15008	106875.0	546.88	1968.75	359.38
20.00%	29489	440015.63	1125.0	6453.13	781.25
30.00%	43704	Más de 20 min	1734.38	No funciona*	1171.88
50.00%	71432	Más de 20 min	3093.78	No funciona*	2125.0
80.00%	111781	Más de 20 min	5062.5	No funciona*	3484.38
100.00%	138150	Más de 20 min	6703.13	No funciona*	4421.88

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo.

- Anotación: Quick sort dejó de funcionar desde el archivo del 30% de los datos, no arroja un mensaje de error ni se demora más de 20 minutos, simplemente el programa se queda unos segundos cargando y y vuelve a aparecer en consola "PS C:\Users\MSI Gaming\Desktop\Universidad\Segundo Semestre\EDA\Reto1-G04> " como si se hubiese cancelado.

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
-small	768	17828.13	875.0	781.25	109.38
5.00%	7572	Más de 20 min	135671.88	223375.0	10625.0
10.00%	15008	Más de 20 min	678890.63	Más de 20 min	42312.5
20.00%	29489	Más de 20 min	Más de 20 min	Más de 20 min	171187.5
30.00%	43704	Más de 20 min	Más de 20 min	Más de 20 min	Más de 20 min
50.00%	71432	Más de 20 min	Más de 20 min	Más de 20 min	Más de 20 min
80.00%	111781	Más de 20 min	Más de 20 min	Más de 20 min	Más de 20 min
100.00%	138150	Más de 20 min	Más de 20 min	Más de 20 min	Más de 20 min

Tabla 3. Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Insertion Sort		
Shell Sort		
Merge Sort	Esta es la opción más rápida	
Quick Sort		

Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

- El ordenamiento más eficiente es Merge sort en la estructura de datos de arreglo. En general todos los ordenamientos son más rápidos en la estructura de arreglo.

Máquina 2

Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAYLIST)	Insertion Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
0.50%					
100.00%					

Tabla 5. Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo.

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
0.50%					
100.00%					

Tabla 6. Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Insertion Sort		
Shell Sort		

Merge Sort		
Quick Sort		

Tabla 7. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

Preguntas de análisis

- 1) ¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?
 - Podemos decir que los algoritmos se comportaron más o menos como se esperaba. Resaltaría la diferencia de que Merge sort y Quick sort se supone que tienen un orden temporal muy parecido en la mayoría de los casos. Sin embargo en este caso Merge sort fue bastante más eficiente y hasta Shell sort fue bastante más eficiente que Quick sort.
- 2) ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?
 - Solo pudimos hacer las pruebas en una máquina.
- 3) De existir diferencias, ¿a qué creen que se deben?
 - Suponemos que habría diferencias dependiendo de las características y capacidades de procesamiento de los equipos.
- 4) ¿Cuál Estructura de Datos funciona mejor si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?
 - La estructura de datos con mejores tiempos en la medición es el ARREGLO.
- 5) Teniendo en cuenta las pruebas de tiempo de ejecución por todos los algoritmos de ordenamiento estudiados (iterativos y recursivos), proponga un ranking de los mismos de mayor eficiencia a menor eficiencia en tiempo para ordenar la mayor cantidad de obras de arte.
 - Merge sort sería el mejor
 - Luego iría Shell sort
 - Luego Quick sort
 - Por último, insertion sort