

OBSERVACIONES DE LA PRACTICA

Estudiante 1 Cod 20172994

Estudiante 2 Cod 201818326

Ambientes de pruebas

	Máquina 1	Máquina 2
Procesadores	Intel Core i7 10750H 2.6Ghz	Intel Core i5 9400F 2.9Ghz
Memoria RAM (GB)	16	16
Sistema Operativo	Windows 10 Home	Windows 10 Home

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

Maquina 1

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%	366	1427	135	130	26
100.00%	736	12203	645	625	83

Tabla 2. 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%	366	1375	125	119	20
100.00%	736	11979	630	640	78

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

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Insertion Sort	12203	11979
Shell Sort	645	630
Merge Sort	625	640
Quick Sort	83	78

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

Maquina 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%	366	93	26	20	0
100.00%	736	749	93	73	15

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%	366	100	20	15	5.2
100.00%	730	750	72	98	15

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

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Insertion Sort	749	750
Shell Sort	93	72
Merge Sort	15	15
Quick Sort	73	98

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?

Recordemos los resultados teóricos de complejidad temporal de los algoritmos de ordenamiento vistos en clase:

Algoritmo	Peor caso	Promedio	Mejor caso
Selection	$O(n^2)$	$O(n^2)$	$O(n^2)$
Insertion	$O(n^2)$	$O(n^2)$	$O(n)$
Shell	$O(n^{3/2})$. *	$O(n^{1.25})$	$O(n \log(n))$
Algoritmo	Peor caso	Promedio	Mejor caso
Quicksort	$O(n^2)$	$O(n \log(n))$	$O(n \log(n))$
MergeSort	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$

Como se puede ver en la tabla anterior, si miramos la columna de complejidad temporal promedio, obtenemos resultados congruentes en la práctica, pues Insertion Sort fue el algoritmo con mayor tiempo de ejecución en ambas máquinas y con ambas TAD (en teoría la complejidad es $O(n^2)$).

Luego, le sigue Shell Sort con una complejidad teórica de $O(n^{1.25})$ que en la práctica fue, como se esperaba, más rápido que Insertion Sort, pero más lento que Merge y Quick Sort, pues estos dos últimos tienen una complejidad temporal promedio teórica de $O(n \log(n))$ que es mejor que $O(n^{1.25})$.

Sin embargo, para Merge y Quick Sort, aunque ambos tienen una complejidad promedio teórica igual, en la práctica se obtuvieron resultados muy superiores con Merge Sort sobre Quick Sort (aproximadamente de 8 a 9 veces más rápido).

- 2) ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?

Sí, la máquina 2 tuvo un menor tiempo de ejecución en todos los casos.

- 3) De existir diferencias, ¿a qué creen que se deben?

Aunque la CPU de la máquina 1 es de una gama superior y de una generación más nueva, el procesador de la máquina 2 tiene una frecuencia de reloj base más alta que la de la máquina 1. Además, se trata de un procesador para escritorio, mientras que el de la máquina 1 es un procesador para laptop. Como es de esperar, los componentes de escritorio tienen un rendimiento muy superior a los de portátil, pues tienen frecuencias más altas y no cuentan con tantas restricciones de temperatura ni energía.

- 4) ¿Cuál Estructura de Datos funciona mejor si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

A nivel general la estructura más eficiente es la LINKED LIST en la mayoría de ordenamientos.

- 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 mismo de mayor eficiencia a menor eficiencia en tiempo para ordenar la mayor cantidad de obras de arte.

De menor a mayor eficiencia esta:

- Selection Sort
- Insertion Sort
- Shell Sort
- Quick Sort
- Merge Sort