

OBSERVACIONES DE LA PRACTICA

Nicolas Merchan Cuestas - 20212109

Ambientes de pruebas

	Máquina 1*	Máquina 2*
Procesadores	Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz 2.11 GHz	Intel(R) Core(TM) i5-3340 CPU @ 2.80GHz 3.10 GHz
Memoria RAM (GB)	16.0 GB	8.0 GB
Sistema Operativo	Windows 10 Enterprise 64-bits	Windows 10 Enterprise 64-bits

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

*En ambos casos se utilizaron equipos facilitados por la universidad. En ese orden de ideas, la presencia de programas de seguridad podría haber afectado el rendimiento del programa.

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]
small	768	243.75	253.13	243.75	281.25
10.00%	13815	25996.88	28825	27746.88	27118.75

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]
small	768	262.5	259.38	246.88	265.63
10.00%	13815	28059.38	29678.13	30856.25	30178.13

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

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Insertion Sort	25996.88	28059.38
Shell Sort	28825	29678.13
Merge Sort	27118.75	30178.13
Quick Sort	27746.88	30856.25

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]
small	768	50	53.13	53.13	53.13
10.00%	13815	5565.625	5543.75	5462.5	5550

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]
small	768	50	59.38	59.38	53.13
10.00%	13815	5612.5	5462.5	5496.88	5550

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

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Insertion Sort	5565.63	5612.5
Shell Sort	5543.75	5462.5
Merge Sort	5550	5550
Quick Sort	5462.5	5496.88

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?

Las experiencias prácticas han sido acordes a los enunciados teóricos. Los tiempos de ejecución correspondientes a los algoritmos de ejecución y las Estructuras de Datos sobre las cuales se ejecutaron son acordes a las proyecciones teóricas hechas antes de la culminación de las pruebas en cuestión. Profundizaremos en la comparación teórico-práctica de los tiempos de procesamiento de cada uno de los algoritmos en el apartado 5 del presente laboratorio.

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

La utilización de maquinas diferentes fue un factor determinante a la hora de realizar las pruebas asignadas. Se presento una diferencia muy considerable entre el tiempo de procesamiento de la máquina 1 y la máquina 2.

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

Considero que la diferencia observada se debe, en parte, a los programas de seguridad instalados en la máquina 1, dado que estos consumen bastante memoria principal y disminuyen la capacidad de procesamiento del equipo. La máquina 1 es un portátil facilitado por la universidad que posee instalados programas para evitar la instalación de programas por parte del estudiante sin autorización del DSIT. Así mismo, el equipo portátil lleva varios años en uso continuo. Por otro lado, la máquina 2

es un equipo de escritorio que, a pesar de poseer programas de seguridad similares, estos no consumen una cantidad semejante de recursos**. Finalmente, es muy importante resaltar que el poder de procesamiento máximo de la máquina 1 es 2,11 GHz mientras aquel de la máquina 2 es 3,10 GHz.

** Comparé el consumo de recursos por dichos programas de seguridad en ambas máquinas.

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

La Estructura de Datos lista ordenada (ARRAY_LIST) tiene un mejor rendimiento general que la lista encadenada (SINGLE_LINKED). Dicho fenómeno se da porque los algoritmos de ordenamiento basan su funcionamiento en gran medida en la búsqueda y comparación de elementos presentes en diferentes posiciones de las Estructuras de Datos. En ese orden de ideas, la búsqueda de elementos en una lista ordenada tiene una complejidad de $O(1)$, mientras que en una lista encadenada dicho proceso tiene una complejidad de $O(n)$. Por lo tanto, la complejidad del programa en el caso de una lista ordenada es menor que aquella en el caso de una lista encadenada.

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 acuerdo con los promedios de tiempo de procesamiento de cada algoritmo teniendo en cuenta ambas máquinas y Estructuras de Datos, se tiene que:

- 1) Insertion Sort (16308 ms)
- 2) Merge Sort (17099 ms)
- 3) Shell Sort (17377 ms)
- 4) Quick Sort (17390 ms)

Los resultados son acordes a las proyecciones teóricas. A pesar de que el algoritmo Insertion Sort tiene una complejidad en el peor caso de $O(n^2)$, en el mejor caso tiene la mejor complejidad entre todos los algoritmos estudiados ($O(n)$). Por lo tanto, se debe dar el caso en el cuál existe un grado de orden en los datos de las obras de arte respecto a su fecha de adquisición antes de ser organizados por los algoritmos. Aquello tiene sentido si se tiene en cuenta que las obras de arte ingresan en la lista de clasificación a medida que son adquiridas por el museo. De la misma manera, el algoritmo Merge Sort cumplió con las expectativas teóricas, dado que en toda situación, su complejidad es $O(n \log(n))$ (la mejor complejidad en el peor caso). Finalmente, los últimos dos algoritmos presentan complejidades muy similares, en concordancia con el hecho de que tanto el algoritmo Shell Sort como el algoritmo Quick Sort poseen complejidades teóricas en el mejor caso del orden de $O(n \log(n))$. Sin embargo, el algoritmo Shell Sort tiene una complejidad mejor ($O(n^{3/2})$) que el algoritmo Quick Sort ($O(n^2)$) en el peor caso.