

OBSERVACIONES DE LA PRACTICA

Hernán Felipe Buitrago Cod 201512807
Daniel Esteban Aguilera Cod 202010592

Máquina 1		Máquina 2
Procesadores	Intel Core i5 Dual-Core @2.60GHz	Intel i5-7400 @3.00GHz
Memoria RAM (GB)	8.0 GB	8.0 GB
Sistema Operativo	MacOS Big Sur	Windows 10 (64- bits)

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

Maquina 1

Resultados

Tamaño de la muestra (ARRAY_LIST)	Insertion Sort (ms)	Selection Sort (ms)	Shell Sort (ms)
1000	920,70	1018,10	56,8
2000	3705,10	4133,50	114,50
4000	14920,10	16633,30	257,50
8000	62626,20	73979,30	643,30
16000	247985,50	300020,50	1489,30
32000	1004073,10	1265941,70	3362,60
64000	Tiempo excesivo	Tiempo excesivo	8542,20
128000	Tiempo excesivo	Tiempo excesivo	19659,90
256000	Tiempo excesivo	Tiempo excesivo	51968,20
512000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo

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

Tamaño de la muestra (LINKED_LIST)	Insertion Sort (ms)	Selection Sort (ms)	Shell Sort (ms)
1000	61158,40	58320,50	3104,50
2000	516290,20	481254,40	13181,70
4000	Tiempo excesivo	Tiempo excesivo	64441,80
8000	Tiempo excesivo	Tiempo excesivo	363617,70
16000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo
32000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo
64000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo
128000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo
256000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo
512000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo

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

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Insertion sort	2	3
Selection sort	3	2
Shell sort	1	1

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

Graficas

- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1**.
 - Comparación de rendimiento ARRAYLIST.
 - Comparación de rendimiento LINKED_LIST.
 - Comparación de rendimiento para Insertion Sort.
 - Comparación de rendimiento para Selection Sort.
 - Comparación de rendimiento para Shell Sort.

Maquina 2

Resultados

Tamaño de la muestra (ARRAY_LIST)	Insertion Sort (ms)	Selection Sort (ms)	Shell Sort (ms)
1000	812,50	937,50	46,88
2000	3375,00	3734,38	93,75
4000	13140,63	15234,38	234,38
8000	55140,63	61859,38	562,50
16000	219593,75	254937,50	1312,50
32000	918296,88	1041187,50	2921,88
64000	3047984,38	4187687,50	7234,38
128000	12926015,63	Tiempo excesivo	17640,63
256000	Tiempo excesivo	Tiempo excesivo	43421,88
512000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo

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

Tamaño de la muestra (LINKED_LIST)	Insertion Sort (ms)	Selection Sort (ms)	Shell Sort (ms)
1000	51156,25	46843,75	2593,75
2000	424796,88	382281,25	10718,75
4000	3395343,75	429125,00	52296,88
8000	Tiempo excesivo	3088515,63	266125,00
16000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo
32000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo
64000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo
128000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo
256000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo
512000	Tiempo excesivo	Tiempo excesivo	Tiempo excesivo

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

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Insertion sort	2	3
Selection sort	3	2
Shell sort	1	1

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

Graficas

- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2**.
 - Comparación de rendimiento ARRAYLIST.
 - Comparación de rendimiento LINKED_LIST.
 - Comparación de rendimiento para Insertion Sort.
 - Comparación de rendimiento para Selection Sort.
 - Comparación de rendimiento para Shell Sort.

Preguntas de análisis

- 1) ¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?
 - Teniendo en cuenta las gráficas es posible observar que el comportamiento de estos ordenamientos va concorde a lo enunciado teóricamente. Aunque es algo difícil de graficar todos los datos tanto en LINKED_LIST como en ARRAY_LIST por la falta de datos ya que es algo complicado tomar todos estos debido al largo tiempo de espera, fue posible evidenciar estos comportamientos.
- 2) ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?
 - Al comparar y observar los tiempos de espera de cada ordenamiento en cada tipo de arreglo podemos concluir que, si hay algunas diferencias, estas se encuentran evidentes en los tiempos de carga, generalmente estos eran más largos en la maquina 1. Aunque se tengan componentes similares, el tiempo de carga siempre va a variar y esto es algo normal ya que cada computador no va a realizar este proceso de forma totalmente igual.
- 3) De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?
 - Estas diferencias se deben a muchos factores. Por un lado, la memoria disponible con la que cuenta el computador es posible ver que ambos cuentan con 8 Gb de RAM, pero no quiere decir que ambos tengan la misma RAM disponible, en muchos casos esta es menor a lo que es indicado en la etiqueta. Por ejemplo, la máquina 2 tiene 8Gb de RAM, pero solo 7,5 utilizable. Por otro lado, el estado de los componentes también afectaría el tiempo de carga. Por último, la cantidad de memoria que estaba siendo utilizada a la hora de realizar las pruebas.
- 4) ¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

- Teniendo en cuenta los tiempos de ejecución en ambos tipos de arreglo, es posible observar que el ARRAY_LIST presenta tiempos de espera menores a los que se observan en el SINGLE_LINKED. Esto se debe a que al estar organizados en un ARRAY_LIST, es mucho más rápido obtener los datos ya que estos se encuentran en orden y junto a los otros. Sin embargo, en la estructura SINGLE_LINKED los datos se encuentran organizados de tal forma que hacen mejor uso de la memoria, esto se puede evidenciar en la memoria usada al cargar los datos, en el caso del ARRAY se usaba un 65% y en el mencionado antes solamente alrededor de 20-30%.