

OBSERVACIONES DE LA PRACTICA

Marilyn Stephany Joven Fonseca-202021346
Samuel Alejandro Jiménez Ramírez-202116652

Ambientes de pruebas

	Máquina 1	Máquina 2
Procesadores	AMD Ryzen 5 4500U, Radeon Graphics 2.38 GHz	Intel(R) Core(TM) i5- 10310U CPU @ 1.70 GHz 2.21GHz
Memoria RAM (GB)	16 GB	16 GB
Sistema Operativo	Windows 10 Home 64-bits	Windows 10 Pro 64- bits

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]
Small	768	2406.25	140.62	156.25	93.75
Small	768	2359.38	125.0	171.88	109.38
Small	768	2312.5	140.62	156.25	109.38
	PROMEDIO	2359.38	135.41	161.46	104.17
10.00%	77	31.25	15.62	15.62	0.0
10.00%	77	46.88	15.62	15.62	15.62
10.00%	77	31.25	15.62	15.62	15.62
	PROMEDIO	36.46	15.62	15.62	10.41

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	16531.25	843.75	859.38	171.88
Small	768	16453.12	843.75	843.75	171.88
Small	768	16453.12	843.75	843.75	171.88
	PROMEDIO	16479.16	843.75	848.96	171.88
10.00%	77	62.5	15.62	15.62	0.0
10.00%	77	62.5	31.25	31.25	15.62
10.00%	77	78.12	31.25	15.62	15.62
	PROMEDIO	67.71	26.04	20.83	10.41

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

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Insertion Sort	2359.38	16479.16
Shell Sort	135.41	843.75

Merge Sort	104.17	171.88
Quick Sort	161.46	848.96

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	3000.0	265.62	328.12	171.88
Small	768	3187.5	234.38	296.88	187.5
Small	768	2953.12	265.62	250.0	218.75
	PROMEDIO	3046,87	255.20	291,66	192,71
10.00%	77	62.6	15.62	31.25	15.62
10.00%	77	46.8	15.62	15.62	15.62
10.00%	77	93.75	15.62	0.0	15.62
	PROMEDIO	67,71	15,62	15,62	15,62

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	24390.62	1328.12	1406.25	343.75
Small	768	24828.12	1328.12	1296.88	312.5
Small	768	24640.62	1328.12	1406.25	328.12
	PROMEDIO	24619,62	1328,12	1369,79	328,12
10.00%	77	125.0	15.62	31.25	31.25
10.00%	77	109.38	31.25	31.25	15.62
10.00%	77	125.0	31.25	31.25	15.62
	PROMEDIO	119,79	26.04	31,25	20.83

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

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Insertion Sort	3046,87	24619,62
Shell Sort	255.20	1328,12
Merge Sort	192,71	328,12
Quick Sort	291,66	1369,79

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?

Teóricamente, el algoritmo que menos se demora y es más eficiente es el merge sort que tiene una complejidad temporal de $n \log(n)$ para el peor y mejor caso, lo que es acorde a lo que vimos reflejado en las pruebas que hicimos al ordenar los datos.

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

Si, existen diferencias respecto al tiempo de ejecución de cualquier algoritmo.

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

Suponemos que debe ser por la tasa de refresco del procesador, o la facilidad que este tiene para procesar este tipo de algoritmos. En nuestro caso, la máquina 1 fue más eficiente.

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

La estructura que mejor comportamiento en términos de tiempo tiene es ARRAY LIST, ya que en ambas maquinas fue la ganadora comparando el tiempo promedio de ordenamiento del 100% de los datos.

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.

Para ambos tipos de lista, el orden sería el siguiente:

1. Merge Sort.
2. Shell Sort.
3. Quick Sort.
4. Insertion Sort.