OBSERVACIONES DE LA PRÁCTICA

Juan Camilo Cancelado 202410123
Gabriela Gomez 202420506
Pedro Archila 202421572

Ambientes de pruebas

	Máquina 1	Máquina 2	Máquina 3
Procesadores	Intel Core i7	Intel Core i5	Chip M1
Memoria RAM (GB)	12GB	8 GB	8G
Sistema Operativo	Windows	Windows	MacOS sequoia 15.1

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

Máquina 1

Resultados para Insertion Sort

Porcentaje de la muestra	Insertion Sort (Array List)	Insertion Sort (Linked List)
0.50%	0.10	7.16
5.00%	120.50	7320.24
10.00%	371.40	33937.32
20.00%	1417.50	MAX RECURSION DEPTH EXCEEDED
30.00%	3090.70	MAX RECURSION DEPTH EXCEEDED
50.00%	7079.88	MAX RECURSION DEPTH EXCEEDED
80.00%	16873.20	MAX RECURSION DEPTH EXCEEDED
100.00%	25922.28	MAX RECURSION DEPTH EXCEEDED

*Tabla 2. **Resultados máquina 1 para insertion Sort***

Resultados para Selection Sort

Porcentaje de la muestra	Selection Sort (Array List)	Selection Sort (Linked List)
0.50%	0.75	3.28
5.00%	88.03	3468.45
10.00%	370.39	28957.67
20.00%	1632.64	MAX RECURSION DEPTH EXCEEDED

30.00%	3461.26	MAX RECURSION DEPTH EXCEEDED
50.00%	9867.51	MAX RECURSION DEPTH EXCEEDED
80.00%	25195.91	MAX RECURSION DEPTH EXCEEDED
100.00%	62373.75	MAX RECURSION DEPTH EXCEEDED

Tabla 3. ***Resultados máquina 1 para Selection Sort***

Resultados para Shell Sort

Porcentaje de la muestra	Shell Sort (Array List)	Shell Sort (Linked List)
0.50%	0.19	1.68
5.00%	3.50	229.76
10.00%	6.58	1470.30
20.00%	13.35	69390.12
30.00%	32.86	MAX RECURSION DEPTH EXCEEDED
50.00%	60.82	MAX RECURSION DEPTH EXCEEDED
80.00%	96.34	MAX RECURSION DEPTH EXCEEDED
100.00%	131.66	MAX RECURSION DEPTH EXCEEDED

Tabla 4. ***Resultados máquina 1 para Shell Sort***

Máquina 2

Resultados para Insertion Sort

Porcentaje de la muestra	Insertion Sort (Array List)	Insertion Sort (Linked List)
0.50% 50	0.50	5.46
5.00% 500	45.30	1328.80
10.00% 1000	143.70	MAX RECURSION DEPTH EXCEEDED
20.00% 2000	506.30	MAX RECURSION DEPTH EXCEEDED
30.00% 3000	1129.40	MAX RECURSION DEPTH EXCEEDED
50.00% 5000	3289.10	MAX RECURSION DEPTH EXCEEDED
80.00% 8000	8799.50	MAX RECURSION DEPTH EXCEEDED
100.00% 10000	14196.90	MAX RECURSION DEPTH EXCEEDED

Tabla 5. ***Resultados máquina 2 para insertion Sort***

Resultados para Selection Sort

Porcentaje de la muestra	Selection Sort (Array List)	Selection Sort (Linked List)

0.50% 50	0.90	4.87
5.00% 500	63.03	1010.22
10.00% 1000	185.02	MAX RECURSION DEPTH EXCEEDED
20.00% 2000	719.76	MAX RECURSION DEPTH EXCEEDED
30.00% 3000	1681.00	MAX RECURSION DEPTH EXCEEDED
50.00% 5000	4562.40	MAX RECURSION DEPTH EXCEEDED
80.00% 8000	17218.98	MAX RECURSION DEPTH EXCEEDED
100.00% 10000	31731.60	MAX RECURSION DEPTH EXCEEDED

Tabla 6. ***Resultados máquina 2 para Selection Sort***

Resultados para Shell Sort

Porcentaje de la muestra	Shell Sort (Array List)	Shell Sort (Linked List)
0.50% 50	0.18	1.35
5.00% 500	3.20	87,13
10.00% 1000	5.20	MAX RECURSION DEPTH EXCEEDED
20.00% 2000	13.18	MAX RECURSION DEPTH EXCEEDED
30.00% 3000	31.62	MAX RECURSION DEPTH EXCEEDED
50.00% 5000	46.60	MAX RECURSION DEPTH EXCEEDED
80.00% 8000	63.63	MAX RECURSION DEPTH EXCEEDED
100.00% 10000	91.96	MAX RECURSION DEPTH EXCEEDED

Tabla 7. ***Resultados máquina 2 para Shell Sort***

Máquina 3

Resultados para Insertion Sort

Porcentaje de la muestra	Insertion Sort (Array List)	Insertion Sort (Linked List)
0.50%	1.84	10.51
5.00%§	47.48	1880.81
10.00%	135.32	15237.11
20.00%	404.43	127330.51
30.00%	900.04	429890.01
50.00%	2460.14	MAX RECURSION DEPTH EXCEEDED
80.00%	6218.51	MAX RECURSION DEPTH EXCEEDED
100.00%	9631.43	MAX RECURSION DEPTH EXCEEDED

Tabla 8. ***Resultados máquina 3 para insertion Sort***

Resultados para Selection Sort

Porcentaje de la muestra	Selection Sort (Array List)	Selection Sort (Linked List)
0.50%	1.45	MAX RECURSION DEPTH EXCEEDED
5.00%	59.68	MAX RECURSION DEPTH EXCEEDED
10.00%	117.49	MAX RECURSION DEPTH EXCEEDED
20.00%	476.73	MAX RECURSION DEPTH EXCEEDED
30.00%	1022.43	MAX RECURSION DEPTH EXCEEDED
50.00%	2769.03	MAX RECURSION DEPTH EXCEEDED
80.00%	7179.56	MAX RECURSION DEPTH EXCEEDED
100.00%	11555.76	MAX RECURSION DEPTH EXCEEDED

Tabla 9. ***Resultados máquina 3 para Selection Sort***

Resultados para Shell Sort

Porcentaje de la muestra	Shell Sort (Array List)	Shell Sort (Linked List)
0.50%	0.40	MAX RECURSION DEPTH EXCEEDED
5.00%	4.86	MAX RECURSION DEPTH EXCEEDED
10.00%	9.18	MAX RECURSION DEPTH EXCEEDED
20.00%	17.04	MAX RECURSION DEPTH EXCEEDED
30.00%	27.54	MAX RECURSION DEPTH EXCEEDED
50.00%	43.38	MAX RECURSION DEPTH EXCEEDED
80.00%	62.44	MAX RECURSION DEPTH EXCEEDED
100.00%	81.57	MAX RECURSION DEPTH EXCEEDED

Tabla 4. ***Resultados máquina 1 para Shell Sort***

Preguntas de análisis

1) **¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?**

   Array_list: En arreglos, la complejidad de los 3 algoritmos de ordenamiento iterativos coincide con su complejidad teórica. Selection e insertion sort tienen un comportamiento cuadrático como se puede evidenciar en la gráfica lo cual coincide con la complejidad teórica de O(N^2). La gráfica

de shell sort tiene un comportamiento no exactamente como el teórico, pero muy similar a la complejidad teórica de O(n^3/2).

Single_linked_list: en listas encadenadas los algoritmos no son sostenibles de ninguna forma, con una muestra pequeña ya se demoran minutos. En el caso de las listas encadenadas la complejidad teórica de insertion y selection es de O(N^3) y en shell sort es de O(N^5/2). Esta complejidad se puede ver perfectamente en lo que se demoran los algoritmos en correr.

2) **¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?**

La mejor estructura de datos para utilizar si únicamente tenemos en cuenta el tiempo que tarda en realizarse son los arreglos, pues en listas encadenadas el proceso de buscar un elemento en la lista y el intercambio de dos elementos tiene una complejidad temporal de O(N) lo que resulta en que la complejidad teórica que se tiene con arreglos se multiplique por N haciendo muy lento el ordenamiento en este tipo de listas.

3) **¿Cómo afecta el tamaño de los datos a la eficiencia de cada algoritmo en las diferentes estructuras de datos?**

Array_list: En el caso de selection sort e insertion sort el aumentar el tamaño de la muestra hace que el tiempo también aumente de forma cuadrática formando una gráfica parabólica como se evidencia. En el caso de shell sort, el tiempo de ejecución aumenta de una forma casi lineal lo que lo hace un algoritmo mucho más rápido y eficiente.

Single_linked_list: En el caso de las single linked lists, con una muestra pequeña ya se demora mucho más que en arreglos y mientras va aumentando el tamaño de la muestra, cada vez se demora muchas veces más respetando la complejidad teórica de cada uno multiplicado por N, pues buscar un elemento e intercambiarlo de puesto con otro tiene una complejidad de N.

4) **¿Cuál consideran que fue el mejor algoritmo iterativo y por qué?**

El mejor algoritmo iterativo tanto teóricamente como en esta práctica fue el shell sort el cual tiene un crecimiento temporal casi lineal pues su complejidad es de O(N^3/2). Además, en cada porcentaje de muestra, el algoritmo que muestra un menor tiempo es el shell sort.

5) **Si quisieras ordenar una lista muy grande de datos, ¿qué algoritmo escogerías? ¿Por qué?**

Para ordenar muchos datos, el mejor algoritmo de ordenamiento es el shell sort gracias a su complejidad temporal, la cual es notablemente menor que la complejidad temporal de selection sort e insertion sort. Sin embargo, si una lista está casi ordenada la mejor opción será el insertion sort, pues en ese caso especial este algoritmo de ordenamiento tiene una complejidad O(N).

6) **Si quisieras ordenar una lista parcialmente ordenada, ¿cuál sería el mejor algoritmo?**

Cuando una lista está parcialmente ordenada es el mejor caso para un insertion sort pues los camios que debe hacer son muy pocos terminando con una complejidad de O(N) convirtiéndose en el mejor de los tres algoritmos de ordenamiento iterativos.