

OBSERVACIONES DE LA PRACTICA

Guillermo Antonio Villalba Escamilla Cod 202114000

Nicolás Ruiz Pérez cod 202123608

Juan José Tovar Ávila Cod 202113204

Maquina 1

Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1149	130,83	145.64	40.32	11,83	10,53
20.00%	4599	1953,29	1937.77	81.34	44,71	48,02
30.00%	6899	4372,46	4239.74	122.64	68,22	67,03
50.00%	11499	11896,32	13553.7	270.70	114,93	112,23
100.00%	22998	50370,8	52229.2	520.64	246,31	241,82

Tabla 1. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1149	5084,83	4238,34	432,65	589,31	87,04
20.00%	4599	335191,67	299751,56	10246,37	12503,26	1258,59
30.00%	6899	1156854,9	1009054,24	25270,79	30699,71	2811.79
50.00%	11499	-	-	107258,58	91638,76	8274,71
100.00%	22998	-	-	4976747,53	406712,97	34031,81

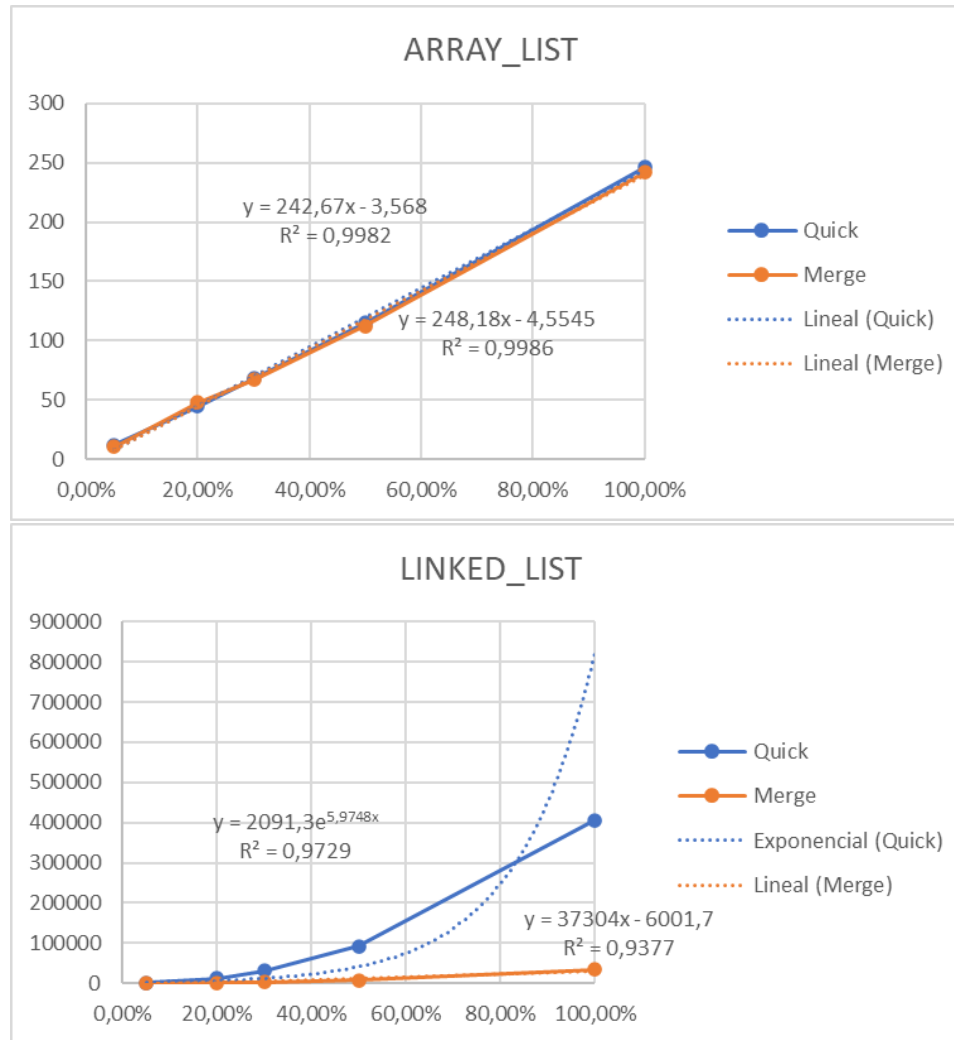
Tabla 2. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	Eficiencia muy similar	X
Quick sort	Eficiencia muy similar	

Tabla 3. 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.
 - Comparación de rendimiento para MergeSort.
 - Comparación de rendimiento para QuickSort.



Maquina 2

Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1149	121,65	189,94	0,058	18,27	18,78
20.00%	4599	1586,78	2985,78	157,98	72,06	70,31
30.00%	6899	5705,44	5769,78	213,56	135,5	101,55
50.00%	11499	15972,23	16671,75	405,96	163,81	167,51
100.00%	22998	64088,43	67489,32	723,75	345,47	341,66

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

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1149	6930,34	5778,66	614,42	883,96	716,01

20.00%	4599	53757,58	49199,12	14000,13	19964,2	1631,3
30.00%	6899	1611147,9	1674780,36	34246,58	43825,3	3731,9
50.00%	11499			107258,58	133408,5	10603,32
100.00%	22998			497647,53	532198,2	50328

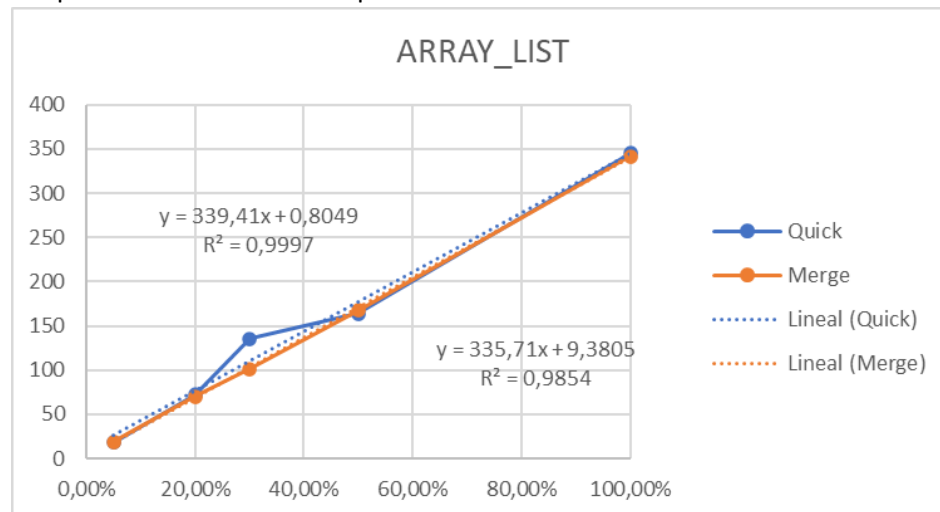
Tabla 6. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

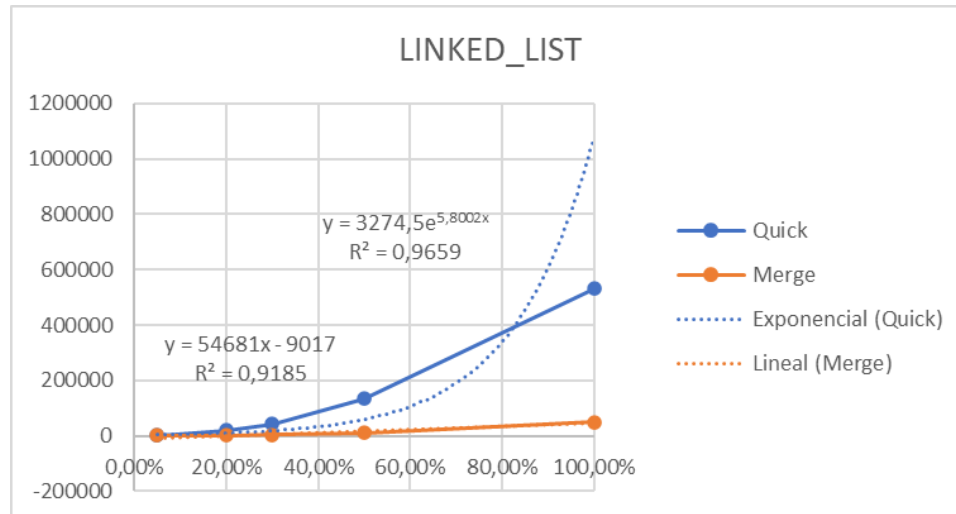
Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	Eficiencia muy similar	X
Quick sort	Eficiencia muy similar	

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.
 - Comparación de rendimiento para MergeSort.
 - Comparación de rendimiento para QuickSort.





Maquina 3

Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1149	100,83	103,06	10,46	15,20	14,02
20.00%	4599	1642,90	1653,22	60,35	64,84	67,08
30.00%	6899	3467,99	3927,82	94,15	99,10	112.56
50.00%	11499	10157,34	10602,03	184,40	208,73	201,34
100.00%	22998	40389,02	43818,06	420,52	403.57	489.17

Tabla 8 Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1149	4599,26	3950,23	394,72	1065,67	182,45
20.00%	4599	313577,80	284466,30	9927,55	28684,54	2842.60
30.00%	6899	-	-	89969,88	67681.66	6680.81
50.00%	11499	-	-	282474,34	194601.94	18818.94
100.00%	22998	-	-	1323713,05	532257	50214,5

Tabla 9. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

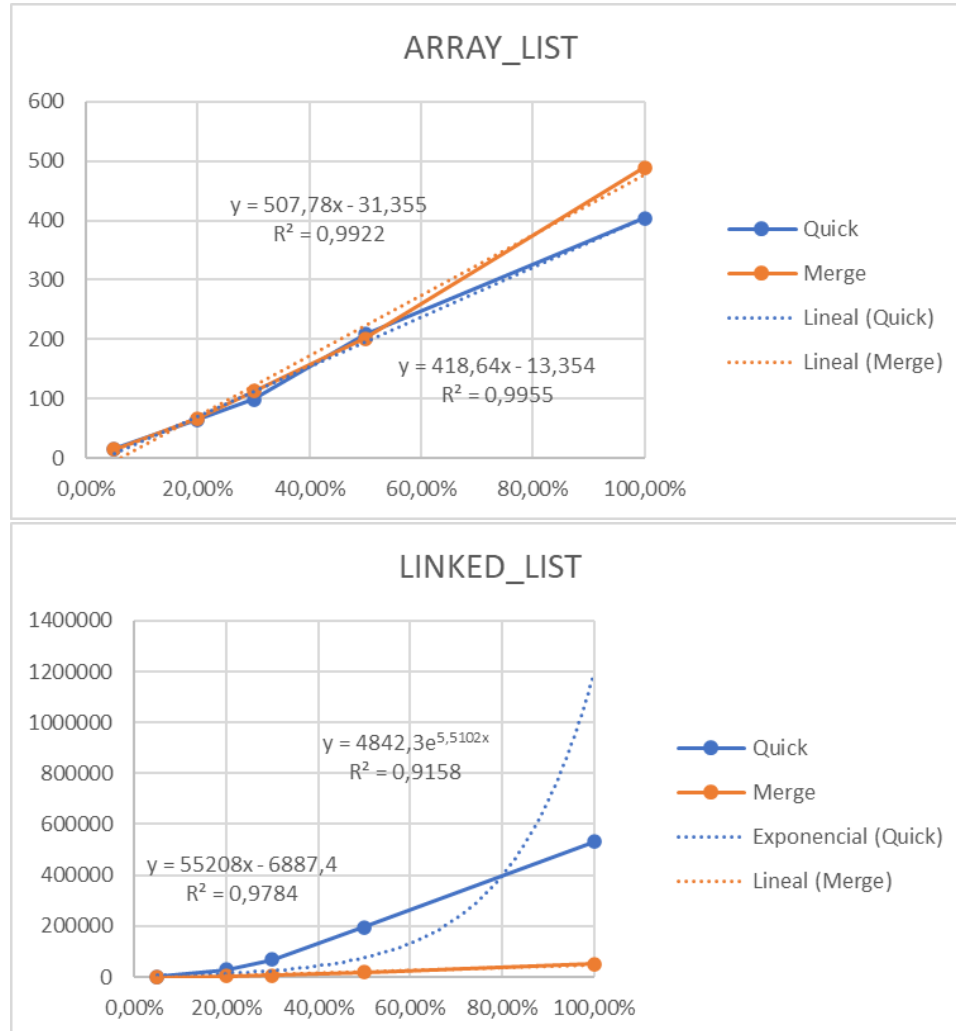
Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	Eficiencia muy similar	X
Quick sort	Eficiencia muy similar	

Tabla 10. 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 3**.
 - 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.
- Comparación de rendimiento para MergeSort.
- Comparación de rendimiento para QuickSort.



Preguntas de análisis

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

Ambas tienen una complejidad similar de orden $O(N \log N)$ en el peor de los casos. La teoría se cumple para listas tipo array, donde quicksort y merge demoran tiempos similares. Hay una pequeña variación al usar datos “-large” donde se concluye que merge es más eficiente para mayores volúmenes de datos.

Por otro lado, la teoría no se cumple para listas tipo linked-list. Las listas de datos más grandes tienen tiempos bastante diferentes donde quicksort se demora bastante en comparación a merge.

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

Sí. Las máquinas con mejores procesadores demoran menos tiempo en ejecutar las pruebas. Si bien los tiempos en las tres máquinas siguieron la misma tendencia, los tiempos de las máquinas con mejores procesadores eran menores que los de las máquinas con los peores procesadores.

3) De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

Esto se debe a que las máquinas con mejores procesadores pueden utilizar más recursos para procesar los datos, lo que hace que se demoren menos. Por otro lado, puede que las máquinas pudieran haber tenido procesos en segundo plano ejecutándose, lo cual reduce los recursos de la máquina para procesar los datos.

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

Array list. Tiene los menores tiempos en todas las calificaciones. Ya que en esta estructura, ambos algoritmos tienen una tendencia lineal.

5) Para el caso analizado de ordenamiento de los artistas, teniendo en cuenta los resultados de tiempo reportados por todos los algoritmos de ordenamiento (iterativos y recursivos), proponga un ranking de los algoritmos de ordenamiento (de mayor eficiencia a menor (en tiempos de ejecución) para ordenar la mayor cantidad de artistas.

1. Merge
2. Quick
3. Shell
4. Insertion
5. Selection