OBSERVACIONES DE LA PRACTICA

Santiago Duque Cod 202021020 (Máquina 1)

Nicolás Guerrero Cod 201731839 (Máquina 2)

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel® Core™ i7-10510U CPU @ 1.80GHz 2.30 GHz | 1.6 GHz Dual-Core Intel Core i5 |
| Memoria RAM (GB) | 16,0 GB | 8 GB 1600 MHz DDR3 |
| Sistema Operativo | Windows 10 Home 64-bits | MacOS Big Sur versión 11.1 |

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 | 796.88 | 578.13 | 31.79 |
| 2000 | 3234.38 | 2265.63 | 62.5 |
| 4000 | 12828.13 | 9125.69 | 156.33 |
| 8000 | 51750.05 | 38125.06 | 343.44 |
| 16000 | 214515.22 | 168593.75 | 812.53 |
| 32000 | 943625.99 | 724437.50 | 1796.21 |
| 64000 | --- | --- | 4390.53 |
| 128000 | --- | --- | 10750.22 |
| 256000 | --- | --- | 26250.16 |
| 512000 | --- | --- | --- |

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 | 30734.38 | 30640.91 | 1848.96 |
| 2000 | 258234.30 | 280906.33 | 7781.25 |
| 4000 | --- | --- | 37382.81 |
| 8000 | --- | --- | 232937.50 |
| 16000 | --- | --- | --- |
| 32000 | --- | --- | --- |
| 64000 | --- | --- | --- |
| 128000 | --- | --- | --- |
| 256000 | --- | --- | --- |
| 512000 | --- | --- | --- |

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 | Mejor | Peor |
| Selection sort | Mejor | Peor |
| Shell sort | Mejor | Peor |

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 | 2282.90 | 2329.52 | 127.95 |
| 2000 | 9284.63 | 10455.545 | 249.82 |
| 4000 | 38088.55 | 38380.29 | 571.78 |
| 8000 | 156003.92 | 155501.46 | 1381.75 |
| 16000 | 631197.53 | 650372.26 | 3346.01 |
| 32000 |  |  | 7176.43 |
| 64000 |  |  | 17734.72 |
| 128000 |  |  | 43265.97 |
| 256000 |  |  | 105838.68 |
| 512000 |  |  |  |

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 | 120362.36 | 125226.19 | 7500.41 |
| 2000 |  | 1011484.406 | 30911.665 |
| 4000 |  |  | 147462.29 |
| 8000 |  |  |  |
| 16000 |  |  |  |
| 32000 |  |  |  |
| 64000 |  |  |  |
| 128000 |  |  |  |
| 256000 |  |  |  |
| 512000 |  |  |  |

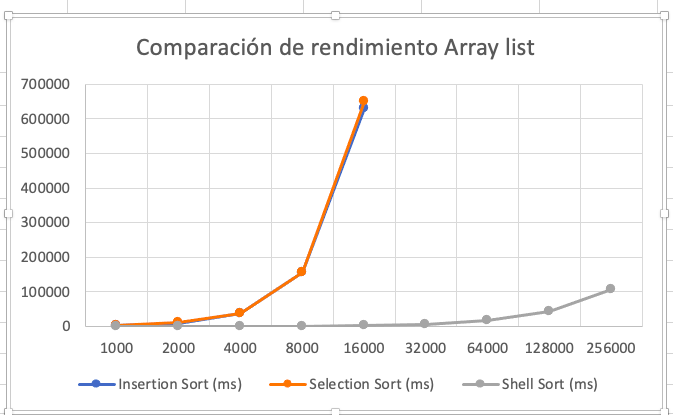
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Insertion sort | Mejor |  |
| Selection sort | Mejor |  |
| Shell sort | Mejor |  |

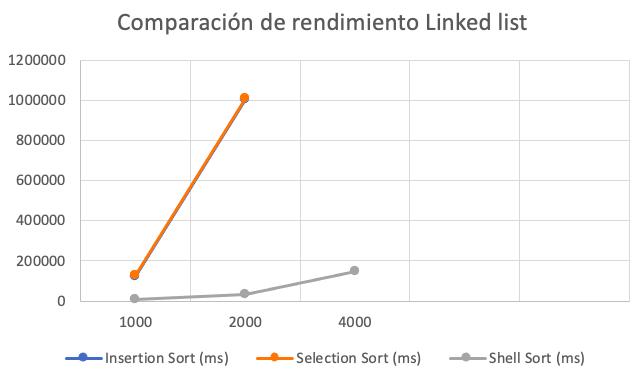
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

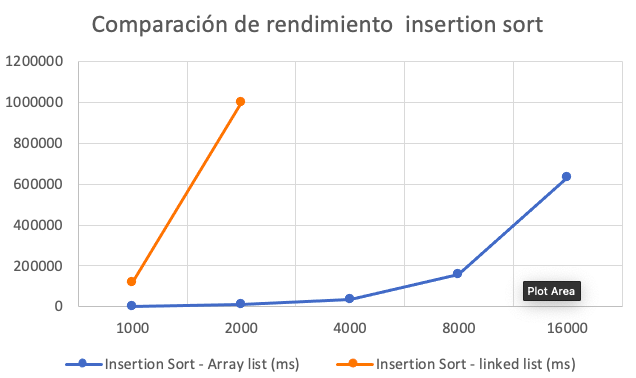
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  + Comparación de rendimiento ARRAYLIST.



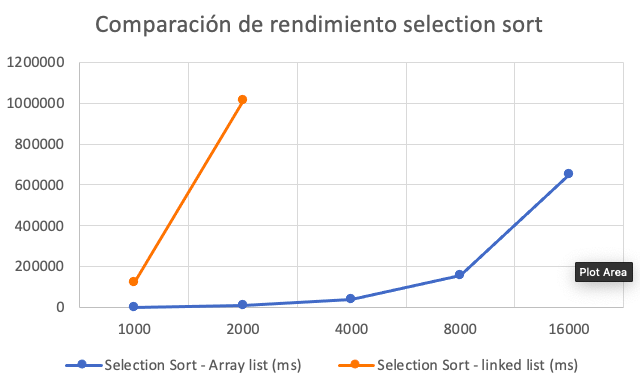
* + Comparación de rendimiento LINKED\_LIST.



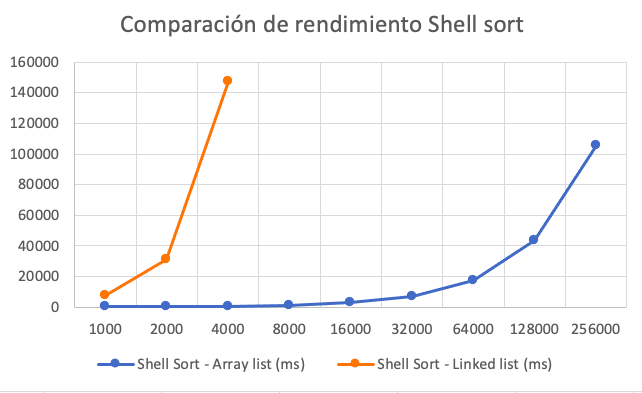
* + Comparación de rendimiento para Insertion Sort.



* + 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?

Como fue previsto en la teoría, el array list resulto más efectivo en cuanto a la manipulación de los datos mediante los algoritmos de ordenamiento. Esto pues al hacer un arreglo de datos, fue posible ordenar con menos de 10 minutos de carga 32000 datos usando selection e insertion sort y 256000 datos al usar el shell sort. Esto comparado con los 2000 datos al usar una lista encadenada con los mismos algoritmos y para el shell sort tan solo 8000. La teoria dice que para un arreglo es más facil realizar algoritmos, que recorran toda la lista, comparado con una lista en cadenada pues en la segunda se debe saltar entre varios lugares en el almacenamiento mientras que en el arreglo todos los datos estan seguidos.

1. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?
2. De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

Esto se puede deber a la diferencia en procesadores u otros factores entre las maquinas, en especial la memoria RAM disponible.

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

Erreglo tuvo mucho mejores tiempos de carga por lo que si solo se tiene en cuenta los tiempos de ejecución, esta es la mejor estructura de datos.