OBSERVACIONES DE LA PRACTICA

Santiago Pardo Morales Cod 202013025

Juan José Ramírez Cala Cod 202013228

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | 2,4 GHz Intel Core i5 de cuatro núcleo | Intel (R) Core (TM) i3-8145U CPU @2.10 GHz 2.30 GHz |
| Memoria RAM (GB) | 8 GB | 4 GB |
| Sistema Operativo | MacOS Catalina | Windows |

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 | 659.66 | 708.78 | 49.68 |
| 2000 | 2766.22 | 2952.30 | 92.53 |
| 4000 | 11596.46 | 12072.17 | 192.60 |
| 8000 | 44995.56 | 54425.11 | 474.38 |
| 16000 | 185267.84 | 236881.24 | 1113.22 |
| 32000 | 820793.35 | Tiempo excedido | 2545.26 |
| 64000 | Tiempo excedido | Tiempo excedido | 6164.48 |
| 128000 | Tiempo excedido | Tiempo excedido | 14872.75 |
| 256000 | Tiempo excedido | Tiempo excedido | 36125.95 |
| 512000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |

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 | 42499.89 | 36960.79 | 2285.80 |
| 2000 | 349214.70 | 313236.85 | 9617.58 |
| 4000 | Tiempo excedido | Tiempo excedido | 45595.77 |
| 8000 | Tiempo excedido | Tiempo excedido | 231993.82 |
| 16000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 32000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 64000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 128000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 256000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 512000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |

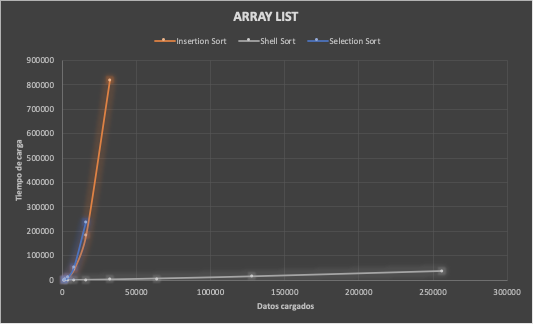
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

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

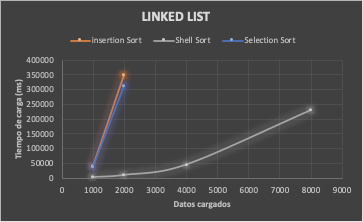
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

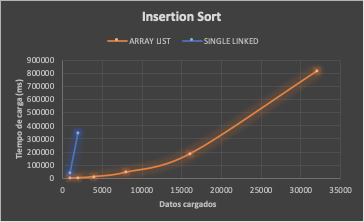
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
  + Comparación de rendimiento ARRAYLIST.



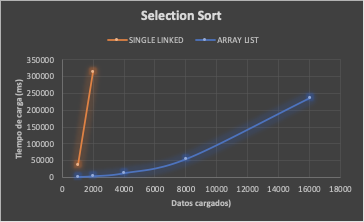
* + Comparación de rendimiento LINKED\_LIST.



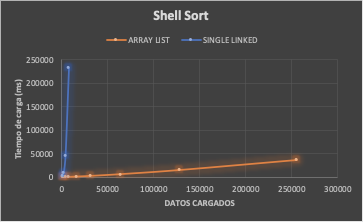
* + Comparación de rendimiento para Insertion Sort.



* + 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 | 953,12 | 1031,25 | 62,5 |
| 2000 | 2828,12 | 3171,87 | 62,5 |
| 4000 | 12046,87 | 13609,37 | 203,12 |
| 8000 | 45859,37 | 60697,5 | 687,5 |
| 16000 | 196890,62 | 239031,25 | 1046,87 |
| 32000 | 843890,62 | 1012796,87 | 3312,5 |
| 64000 | Tiempo excedido | Tiempo excedido | 6203,12 |
| 128000 | Tiempo excedido | Tiempo excedido | 16187,5 |
| 256000 | Tiempo excedido | Tiempo excedido | 46546,87 |
| 512000 | Tiempo excedido | Tiempo excedido | No calculado |

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

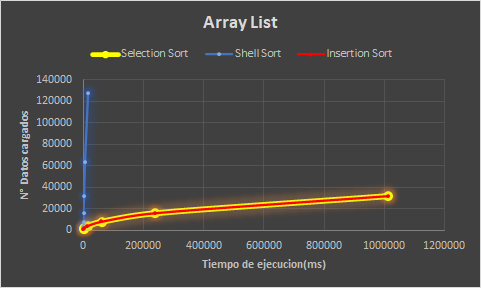
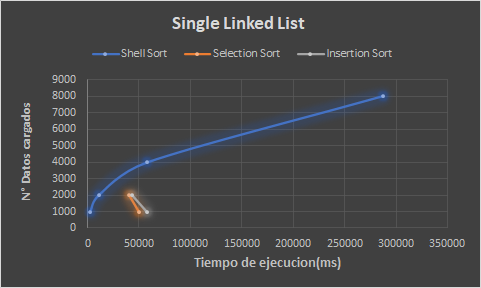
|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño de la muestra (LINKED\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) |
| 1000 | 58062,5 | 49937,5 | 2625 |
| 2000 | 42762,5 | 407718,75 | 10875 |
| 4000 | Tiempo excedido | Tiempo excedido | 57875 |
| 8000 | Tiempo excedido | Tiempo excedido | 286750 |
| 16000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 32000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 64000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 128000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 256000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 512000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |

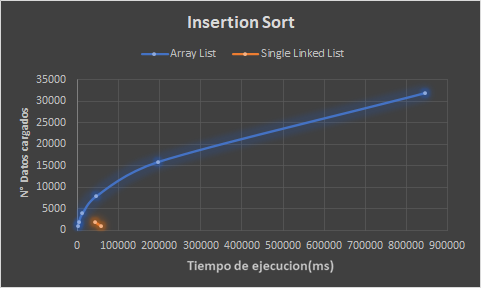
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

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

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

## **Graficas**

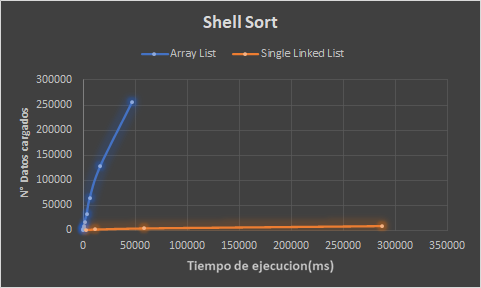
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  + Comparación de rendimiento ARRAYLIST.
  + 
  + Comparación de rendimiento LINKED\_LIST.
  + 
  + Comparación de rendimiento para Insertion Sort.



* + 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?

Si, de acuerdo a los datos reunidos durante las pruebas se pueve evidenciar que los algoritmos con mayor complejidad (insertion y selection sort) tomaron mas tiempo de ejecución que los de menor complejidad (shell sort)

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

Hubo una ligera diferencia en milisegundos con el tiempo de carga de los datos, favoreciendo a la máquina 1. Sin embargo, no fue tan notorio el cambio y el comportamiento de las gráficas fue similar

1. De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

Pudieron existir dos razones muy importantes

La primera tiene que ver con el procesador y la memoria RAM de la máquina 1 con respecto a la máquina 2, por lo cual la máquina 1 se ve favorecida y esto se puede observar en los resultados.

La segunda tiene que ver con las condiciones en las que se realizó el laboratorio, es decir, qué tan desocupado estaba el computador estaba para poder cargar estos datos. Tiene que ver si las aplicaciones y demás servicios, aparte de Visual Studio, estaban cerradas.

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

El ARRAY-LIST de lejos, es la mejor estructura de datos para manipular esta ejecución, ya que su estructura permite un acceso fácil, permitiéndola ordenar de una manera más rápida. Esto se puede evidenciar con las comparaciones entre algoritmos para cada tipo de lista, en los cuales las complejidades de las funciones pertenecientes a este tipo de estructura son generalmente favorables, por ejemplo, en insertion y selection sort se puede ver una complejidad cercana a una función NlogN y en shell sort un comportamiento similar a LogN