OBSERVACIONES DE LA PRACTICA

Juan Camilo Falla 201922219

Nicolás Klopstock 202021352

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 (M.1) | Máquina 2 (M.2) |
| Procesadores | i7 10700 @2.9GHz- 4.8GHz | i5 6200U CPU @2.30GHz 2.40GHz |
| Memoria RAM (GB) | 16 gb | 4 gb |
| Sistema Operativo | Windows 10 Home | Windows 10 Home |

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 | 343.75 | 359.38 | 15.63 |
| 2000 | 1437.5 | 1515.25 | 31.25 |
| 4000 | 5890.63 | 6187.5 | 93.75 |
| 8000 | 23531.25 | 24406.25 | 234.38 |
| 16000 | 94984.38 | 102109.38 | 546.88 |
| 32000 | 390687.5 | 429140.63 | 1343.75 |
| 64000 | Excede 12 mins | Excede 12 mins | 3125 |
| 128000 | Excede 12 mins | Excede 12 mins | 7781.25 |
| 256000 | Excede 12 mins | Excede 12 mins | 18406.25 |
| 512000 | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. |

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 | 26750 | 24546.88 | 1234.38 |
| 2000 | 230921.88 | 205937.5 | 6921.88 |
| 4000 | Excede 12 mins | Excede 12 mins | 333296.88 |
| 8000 | Excede 12 mins | Excede 12 mins | 146843.75 |
| 16000 | Excede 12 mins | Excede 12 mins | 698015.63 |
| 32000 | Excede 12 mins | Excede 12 mins | Excede 12 mins |
| 64000 | Excede 12 mins | Excede 12 mins | Excede 12 mins |
| 128000 | Excede 12 mins | Excede 12 mins | Excede 12 mins |
| 256000 | Excede 12 mins | Excede 12 mins | Excede 12 mins |
| 512000 | Excede 12 mins | Excede 12 mins | Excede 12 mins |

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 | X |  |
| Selection sort | X |  |
| Shell sort | 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 | 703,125 | 781,25 | 46,875 |
| 2000 | 3171,875 | 3656,25 | 93,75 |
| 4000 | 12750 | 12843,75 | 250 |
| 8000 | 52203,125 | 53015,625 | 531,5 |
| 16000 | 211265,625 | 221625 | 1218,75 |
| 32000 | excede 20 mins | excede 20 mins | excede 20 mins |
| 64000 | excede 20 mins | excede 20 mins | excede 20 mins |
| 128000 | excede 20 mins | excede 20 mins | excede 20 mins |
| 256000 | excede 20 mins | excede 20 mins | excede 20 mins |
| 512000 | excede 20 mins | excede 20 mins | excede 20 mins |

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 | 68890.625 | 69296.875 | 3109.375 |
| 2000 | 9984375999 | 499843,75 | 16281,25 |
| 4000 | excede 20 mins | excede 20 mins | 82671,875 |
| 8000 | excede 20 mins | excede 20 mins | 360484,375 |
| 16000 | excede 20 mins | excede 20 mins | excede 20 mins |
| 32000 | excede 20 mins | excede 20 mins | excede 20 mins |
| 64000 | excede 20 mins | excede 20 mins | excede 20 mins |
| 128000 | excede 20 mins | excede 20 mins | excede 20 mins |
| 256000 | excede 20 mins | excede 20 mins | excede 20 mins |
| 512000 | excede 20 mins | excede 20 mins | excede 20 mins |

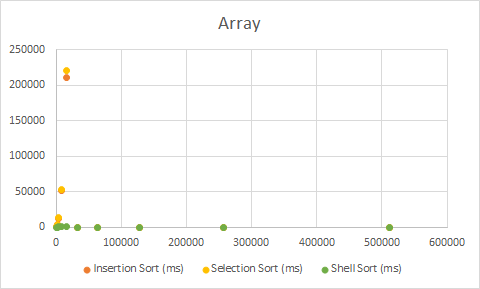
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

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

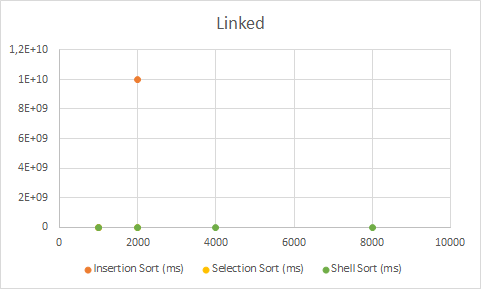
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

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

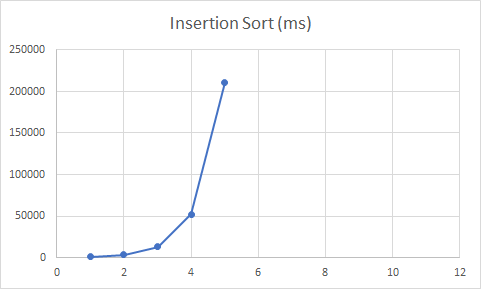


* + Comparación de rendimiento LINKED\_LIST.

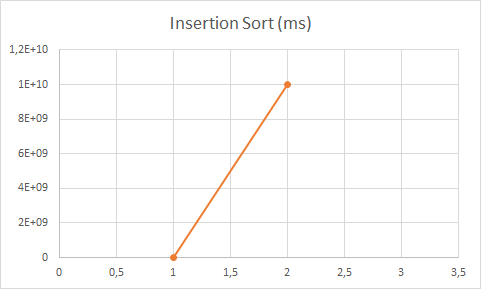


* + Comparación de rendimiento para Insertion Sort.

ARRAY:

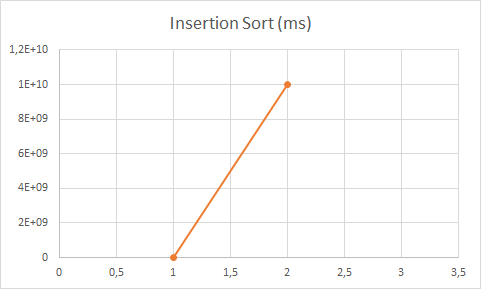


LINKED:

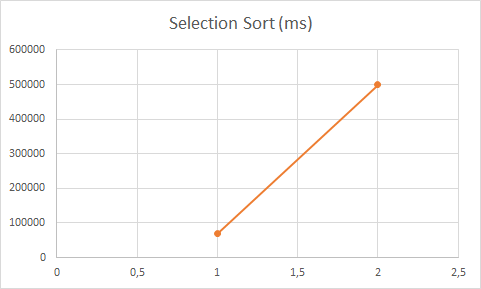


* + Comparación de rendimiento para Selection Sort.

ARRAY:

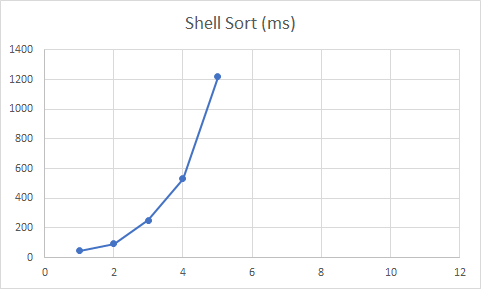


LINKED:

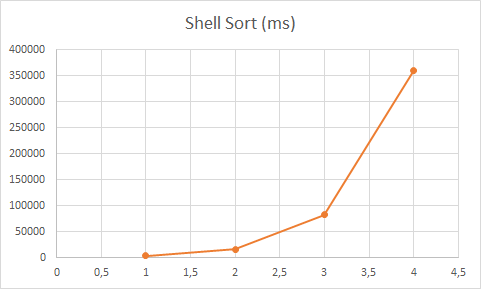


* + Comparación de rendimiento para Shell Sort.

ARRAY:



LINKED:



# **Preguntas de análisis**

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

Los comportamientos de los algoritmos son de carácter potencial y se demuestra en las pruebas realizadas. Sin embargo, no siempre coinciden de manera precisa y esto se debe al tipo de estructura utilizada para cargar y almacenar los datos del archivo. Era evidente al usar las listas enlazadas que los algoritmos de ordenamiento tardaban más tiempo en terminar sus procesos y solo podían organizar 2000 datos antes de que se excediera el límite de tiempo (12 mins). Esto se ve en las tendencias de las gráficas, las cuales demuestran un comportamiento esperado y muy cercano a la teoría cuando se organizan arreglos. Pero este comportamiento se aleja de la teoría vista en clase al momento de utilizar listas enlazadas.

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

Al usar dos máquinas para realizar las mismas pruebas, se observa una diferencia significativa en los resultados de estas. Muchas de las pruebas tienen una alta diferencia en cuanto al tiempo de cada función de ordenamiento y esto lleva a que haya variedades en la rapidez de las funciones dependiendo del tipo de lista que se cargó.

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

Pensamos que se debe a las diferecias entre el procesador de una máquina y la otra. Además de esto, pensamos que las gigas de memoria *RAM* también afectan en este caso y se puede ver el cambio de tiempo entre una y otra al evaluar las funciones.

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

Según los resultados, después de evaluar cada función de sort en los dos tipos de estructura de datos (ARRAY\_LIST y LINKED\_LIST), podemos concluir que, en nuestro caso, es más rápido usar el tipo de lista ARRAY\_LIST. Esto era lo que esperábamos, ya que un array, al ser lineal y tener los elementos “uno al lado del otro”, más fácil encontrar un dato y ordenarlo.