OBSERVACIONES DE LA PRACTICA

Cristian Armando Sánchez Ocampo 202022112

Luis Felipe Dussán R- 201912309

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD Ryzen 3 2200G 3.85 GHz | 1,4 GHz Quad-Core Intel Core i5 |
| Memoria RAM (GB) | 8.00GB | 8 GB 2133 MHz LPDDR3 |
| Sistema Operativo | Windows 10Pro 64 bits | Mac.OS Big Sur. Versión 11.2.1 |

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

# Maquina 1

Límite de ms de carga: 500,000.00 ms

## Resultados

|  |  |  |  |
| --- | --- | --- | --- |
| **Tamaño de la muestra (ARRAY\_LIST)** | **Insertion Sort (ms)** | **Selection Sort (ms)** | **Shell Sort (ms)** |
| **1000** | 41,781.25 | 38,046.88 | 2,093.75 |
| **2000** | 386,218.75 | 339,734.38 | 10,968.75 |
| **4000** | 3,087,687.50 | 2,673,890.62 | 53,484.00 |
| **8000** |  |  | 247,468.75 |
| **16000** |  |  |  |
| **32000** |  |  |  |
| **64000** |  |  |  |
| **128000** |  |  |  |
| **256000** |  |  |  |
| **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** | 43,906.25 | 41,031.25 | 2,234.38 |
| **2000** | 383,125.00 | 333,609.38 | 11,046.88 |
| **4000** | Fuera del límite | Fuera del límite | 54,265.62 |
| **8000** |  |  | 276,671.88 |
| **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** | 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.
* Chart, line chart

  Description automatically generated
* Comparación de rendimiento LINKED\_LIST. Chart, line chart

  Description automatically generated
* Comparación de rendimiento para Insertion Sort.

Chart, line chart

Description automatically generated

* Comparación de rendimiento para Selection Sort.

Chart, line chart

Description automatically generated

* Comparación de rendimiento para Shell Sort.

Chart, line chart

Description automatically generated

# **Maquina 2**

## **Resultados**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) |
| 1000 | 41741.8 ms | 38450.7 ms | 2087.15 ms |
| 2000 | 347035.58 ms | 312248.34 ms | 9893.49 ms |
| 4000 | 2833975.08 ms | Sobrepasó el límite | 47825.52 ms |
| 8000 | Sobrepasó el límite | Sobrepasó el límite | 226429.79 ms |
| 16000 | Sobrepasó el límite | Sobrepasó el límite | 1004664.4 ms |
| 32000 | Sobrepasó el límite | Sobrepasó el límite | Sobrepasó el límite |
| 64000 | Sobrepasó el límite | Sobrepasó el límite | Sobrepasó el límite |
| 128000 | Sobrepasó el límite | Sobrepasó el límite | Sobrepasó el límite |
| 256000 | Sobrepasó el límite | Sobrepasó el límite | Sobrepasó el límite |
| 512000 | Sobrepasó el límite | Sobrepasó el límite | Sobrepasó el límite |

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 | 41982.65 ms | 37588.97 ms | 2085.63 ms |
| 2000 | 344723.33 ms | 305610.12 ms | 9755.4 ms |
| 4000 | Sobrepasó el límite | Sobrepasó el límite | 45157.44 ms |
| 8000 | Sobrepasó el límite | Sobrepasó el límite | 226192.09 ms |
| 16000 | Sobrepasó el límite | Sobrepasó el límite | 1033079.11 ms |
| 32000 | Sobrepasó el límite | Sobrepasó el límite | Sobrepasó el límite |
| 64000 | Sobrepasó el límite | Sobrepasó el límite | Sobrepasó el límite |
| 128000 | Sobrepasó el límite | Sobrepasó el límite | Sobrepasó el límite |
| 256000 | Sobrepasó el límite | Sobrepasó el límite | Sobrepasó el límite |
| 512000 | Sobrepasó el límite | Sobrepasó el límite | Sobrepasó el límite |

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 | No | Sí |
| Selection sort | Sí | No |
| Shell sort | No | Sí |

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?

No, en tería el ARRAY\_LIST es más eficiente. Sin embargo, en la teoria nos arrojan resultados diferentes. El más eficiente parece ser el Linked\_List en mi ordenador. Solo en el caso de Selection Sort de la maquina 2, fue más eficiente ARRAY\_LIST, pero por muy poco.

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

Aunque por obvias razones ambas maquinas no van a obtener los mismos datos, la tendencia está parecida, si existe una similitud entre ambos ordenadores. Sin embargo, hay algunos datos donde la maquina dos se demora menos en hacer ciertas operaciones que la maquina 1. Como por ejemplo cuando queremos cargar 4000 datos en ARRAY, con la estructura de datos Insertion Sort.

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

Estas diferencias pequeñas pueden ser a causa de las diferencias en los procesadores de ambas maquinas, incluso en el código.

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

La mejor opción claramente es Shell Sort(), ordena de manera más eficiente los datos en todos los casos.