OBSERVACIONES DE LA PRACTICA

* E1: Daniela [Alvarez-202020209-d.alvarezr@uniandes.edu.co](mailto:Alvarez-202020209-d.alvarezr@uniandes.edu.co)
* E2: Andres [Mendoza-202012676-af.mendoza@uniandes.edu.co](mailto:Mendoza-202012676-af.mendoza@uniandes.edu.co)

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD Ryzen 7 3700 with Radeon Vega Mobile Gfx 2.30GHz | AMD Ryzen 7 3300 |
| Memoria RAM (GB) | 8,00 Gb | 12,00 Gb |
| Sistema Operativo | Windows 10 64bits | Windows 10 64bits |

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 | 1359.38 | 1171.88 | 78.13 |
| 2000 | 5437.50 | 4765.63 | 171.88 |
| 4000 | 21359.38 | 20093.75 | 375.00 |
| 8000 | 94406.25 | 91031.25 | 953.13 |
| 16000 | 385328.13 | 398500.0 | 2187.5 |
| 32000 | Tiempo excesivo > 15min | Tiempo excesivo > 15min | 5234.38 |
| 64000 | Tiempo excesivo > 15min | Tiempo excesivo > 15min | 12000.0 |
| 128000 | Tiempo excesivo > 15min | Tiempo excesivo > 15min | 28640.63 |
| 256000 | Tiempo excesivo > 15min | Tiempo excesivo > 15min | 65656.25 |
| 512000 | Muestra mayor al tamaño de la lista | | |

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 | 73796.88 | 63109.38 | 3812.5 |
| 2000 | 782437.5 | 535343.75 | 17000.0 |
| 4000 | Tiempo excesivo > 15min | Tiempo excesivo > 15min | 72515.63 |
| 8000 | Tiempo excesivo > 15min | Tiempo excesivo > 15min | 377781.25 |
| 16000 | Tiempo excesivo > 15min | Tiempo excesivo > 15min | Tiempo excesivo > 15min |
| 32000 | Tiempo excesivo > 15min | Tiempo excesivo > 15min | Tiempo excesivo > 15min |
| 64000 | Tiempo excesivo > 15min | Tiempo excesivo > 15min | Tiempo excesivo > 15min |
| 128000 | Tiempo excesivo > 15min | Tiempo excesivo > 15min | Tiempo excesivo > 15min |
| 256000 | Tiempo excesivo > 15min | Tiempo excesivo > 15min | Tiempo excesivo > 15min |
| 512000 | Muestra mayor al tamaño de la lista | | |

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** (Según los datos obtenidos en las pruebas de rendimiento, es mas efectivo en esta estructura, sin embargo, en la regresión que se obtuvo de las gráficas es mas efectivo en la otra. Lo anterior puede explicarse debido a la inexactitud que implica cualquier regresión y la falta de registros para LINKED-LIST debido al exceso de tiempo en el procesamiento de los datos) |  |
| Selection sort | **X** (Según los datos, es mas efectiva en este tipo de lista. Sin embargo, también debido a la regresión, en la gráfica aparece que después de cierta cantidad de datos LINKED LIST es mas efectiva. |  |
| Shell sort | **X** (Tanto la gráfica como los datos confirman que es mas efectiva acá) |  |

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 |  |  |  |
| 2000 |  |  |  |
| 4000 |  |  |  |
| 8000 |  |  |  |
| 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 |  |  |  |
| 2000 |  |  |  |
| 4000 |  |  |  |
| 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 |  |  |
| Selection sort |  |  |
| Shell sort |  |  |

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, ya que en efecto el crecimiento temporal de insertion sort y selection sort resultó ser mucho mayor al de shell sort. Adiconalmente, este último se acopló a una gráfica logarítmica, que coincide con su orden de crecimiento. Para comprobar el orden de crecimiento cuadrático de los primeros dos métodos de ordenamiento seria necesario tomar sus tiempos exactos, sin embargo, estas se ajustaron a una función polinomial y se pudo comprobar la diferencia de tiempo con el otro método.

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

Si, los tiempos de ejecución variaron ligeramente

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

Consideramos que estas diferencias pueden ser a causa del procesador de cada equipo, que pueden ejecutar los ordenamientos en un tiempo mayor o menor.

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

La de shell sort. Lo anterior debido a que el tiempo de ejecución del ordenamiento es mucho menor.