OBSERVACIONES DE LA PRACTICA

Nicolas Maldonado Cod 201921739

David Rincon Cod 201921719

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz | Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz |
| Memoria RAM (GB) | 16GB | 8,00 GB |
| Sistema Operativo | Windows 10 64-bit | Windows 10 64-bit |

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 | 0.0 | 640.62 | 0.0 |
| 2000 | 15.62 | 2640.62 | 62.50 |
| 4000 | 8734.37 | 10656.24 | 140.62 |
| 8000 | 36421.87 | 45906.24 | 109.37 |
| 16000 | 146765.62 | 198843.75 | 218.75 |
| 32000 | +30min | +30min | 437,5 |
| 64000 | +30min | +30min | 4671.87 |
| 128000 | +30min | +30min | 11203,12 |
| 256000 | +30min | +30min | 29265.62 |
| 512000 | SUPERO TAMAÑO DE LISTA | SUPERO TAMAÑO DE LISTA | SUPERO TAMAÑO DE 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 | 32250.00 | 24359.37 | 281,25 |
| 2000 | 264531.25 | 199343.75 | 1234.37 |
| 4000 | 638436.76 | 543867.76 | 34406.25 |
| 8000 | 1487326.34 | 1362864.65 | 12109.37 |
| 16000 | 3867465.34 | 3437394.34 | 17453.12 |
| 32000 | +30min | +30min | +30min |
| 64000 | +30min | +30min | +30min |
| 128000 | +30min | +30min | +30min |
| 256000 | +30min | +30min | +30min |
| 512000 | SUPERO TAMAÑO DE LISTA | SUPERO TAMAÑO DE LISTA | SUPERO TAMAÑO DE 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 |  |
| 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 | 0.0 | 843.62 | 0.0 |
| 2000 | 19.74 | 4756.43 | 65.50 |
| 4000 | 10056.56 | 15656.24 | 133.63 |
| 8000 | 47843.28 | 68984.62 | 140.45 |
| 16000 | 194367.74 | 211856.74 | 274.74 |
| 32000 | +30min | +30min | 532.45 |
| 64000 | +30min | +30min | 5432.63 |
| 128000 | +30min | +30min | 13376.74 |
| 256000 | +30min | +30min | 32476.63 |
| 512000 | SUPERO TAMAÑO DE LISTA | SUPERO TAMAÑO DE LISTA | SUPERO TAMAÑO DE 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 | 35240.00 | 34359.37 | 381,25 |
| 2000 | 289521.25 | 329313.75 | 1454.43 |
| 4000 | 632746.64 | 843957.75 | 38068.35 |
| 8000 | 1452846.26 | 2012486.37 | 29674.32 |
| 16000 | 3276395.45 | 4328495.13 | 27304.27 |
| 32000 | +30min | +30min | +30min |
| 64000 | +30min | +30min | +30min |
| 128000 | +30min | +30min | +30min |
| 256000 | +30min | +30min | +30min |
| 512000 | SUPERO TAMAÑO DE LISTA | SUPERO TAMAÑO DE LISTA | SUPERO TAMAÑO DE 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 |  |
| 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.**

**Clarificación:** Desde el tamaño 32000, la mayoría de las pruebas tardaban más de 30 minutos, por lo que esto se verá en la gráfica como una constante, tampoco se graficó hasta el punto 512000 ya que esta constante hubiera vuelto los datos más difíciles de visualizar (debido a que no se tenía la duración exacta por su largo tiempo de procesamiento).

# **Preguntas de análisis**

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

-Si, cada uno de los algoritmos estudiados generó resultados acorde a lo que se esperaba. De esta manera, shell sort cumplió con las espectativas de ser el algoritmo mas eficiente en cuanto al tiempo de ejecución. Por el otro lado, insertion sort y selection sort al tener listas de elementos muy grandes mucho más tiempo que shell sort.

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

-Al tratarse de un programa el cual se basa en el ordenamiento de elementos en memoria, es muy importante tener en cuenta que cada maquina procesa los datos a una velocidad dependiente del procesador y aunque en el caso de las dos máquinas usadas en este caso la diferencia es mínima, los resultados al ejecutar pruebas de ordenamiento de datos en las maquinas siempre serán diferentes.

3) De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

-La razón principal por la cual existen diferencias en los resultados se debe al rendimiento de procesamiento de cada máquina. De esta manera, una máquina con un mejor procesador mostrará mejores resultados en el tiempo de de ejecución.

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

-Shell sort es la mejor estructura de datos al tener en cuenta solamente el tiempo de ejecución.