OBSERVACIONES DE LA PRACTICA

Estudiante 1: Juan Sebastián Ortega Romero | Cod 202021703

Estudiante 2: Yesid Camilo Almanza | Cod 201921773

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Procesador Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz | Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz |
| Memoria RAM (GB) | 16GB | 16GB |
| Sistema Operativo | Windows 10 Home 64-bits | Windows 10 Home 64-bits |

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

De acuerdo a lo mencionado en clase magistral, el tiempo de ejecución máximo que se tomará en cuenta será de **12 minutos**. Sin embargo, con el objetivo de ampliar nuestro marco de referencia en el momento de crear las gráficas, algunos datos se tomaron a pesar de que dicho tiempo máximo fue superado considerablemente.

En los casos en los que el tiempo máximo fue superado (excediento la capacidad efectiva de procesamiento de nuestros terminales) no se realizó un registro. Estos caso fueron consignados en la tabla como campos NAN.

# **Maquina 1**

## **Resultados**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) |
| 1000 | 67125.0 | 51562.5 | 2796.88 |
| 2000 | 438984.38 | 323828.13 | 14078.13 |
| 4000 | 3627187.5 | 2653000.0 | 61109.34 |
| 8000 | NAN | 23762031.25 | 300484.38 |
| 16000 | NAN | NAN | 1403406.25 |
| 32000 | NAN | NAN | 6511781.25 |
| 64000 | NAN | NAN | 34152906.25 |
| 128000 | NAN | NAN | NAN |
| 256000 | NAN | NAN | NAN |
| 512000 | NAN | NAN | NAN |

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 | 60453.13 | 55843.75 | 2734.375 |
| 2000 | 471734.38 | 444406.25 | 13265.625 |
| 4000 | 4250890.625 | 3190515.625 | 62093.75 |
| 8000 | NAN | NAN | 334156.25 |
| 16000 | NAN | NAN | NAN |
| 32000 | NAN | NAN | NAN |
| 64000 | NAN | NAN | NAN |
| 128000 | NAN | NAN | NAN |
| 256000 | NAN | NAN | NAN |
| 512000 | NAN | NAN | NAN |

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 | 77% | 73.4% |
| Selection sort | 83.4% | 80% |
| Shell sort | 74.78% | 87.81 |

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 | 30750,0 | 34562,5 | 1828,125 |
| 2000 | 279062,5 | 246703,13 | 8828,125 |
| 4000 | 2326093,75 | 2063234,38 | 38625,0 |
| 8000 | NAN | NAN | 194500,0 |
| 16000 | NAN | NAN | 830093,75 |
| 32000 | NAN | NAN | 4071296,875 |
| 64000 | NAN | NAN | NAN |
| 128000 | NAN | NAN | NAN |
| 256000 | NAN | NAN | NAN |
| 512000 | NAN | NAN | NAN |

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 | 30468,75 | 35640,625 | 1734,375 |
| 2000 | 277578,13 | 249796,88 | 8109,38 |
| 4000 | 2300375,0 | 2125640,63 | 36843,8 |
| 8000 | NAN | NAN | NAN |
| 16000 | NAN | NAN | NAN |
| 32000 | NAN | NAN | NAN |
| 64000 | NAN | NAN | NAN |
| 128000 | NAN | NAN | NAN |
| 256000 | NAN | NAN | NAN |
| 512000 | NAN | NAN | NAN |

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 | 85.4619% | 85.622% |
| Selection sort | 87.1047% | 86.714% |
| Shell sort | 19.30115% | 23.02259% |

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

## **Graficas**

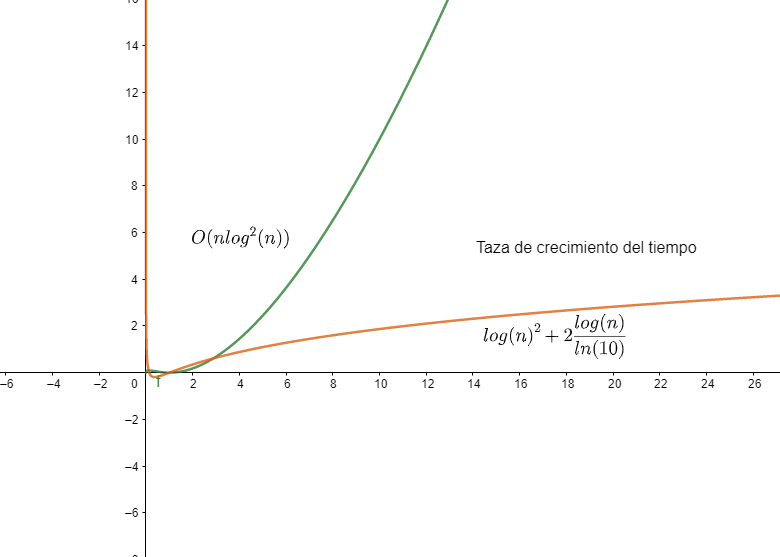
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  + Comparación de rendimiento ARRAYLIST.
  + Comparación de rendimiento LINKED\_LIST.
  + Comparación de rendimiento para Insertion Sort.
  + Comparación de rendimiento para Selection Sort.
  + Comparación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

# **Preguntas de análisis**

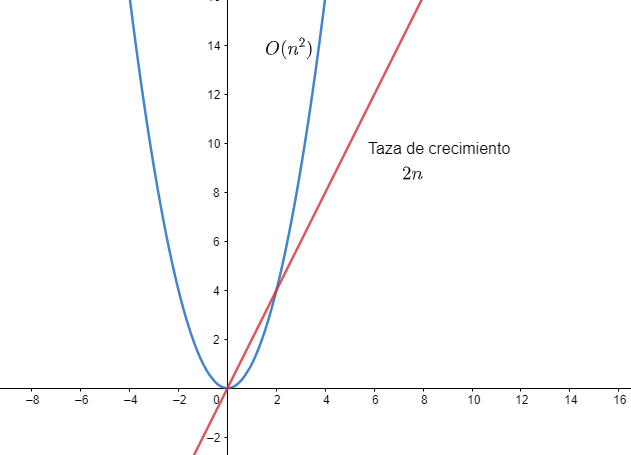
1. ¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?

Los algoritmos Selection Sort e Insertion Sort cuentan con una complejidad temporal qué, en el peor de los casos, es de O(n2). Por otro lado, el algorítmo Shell Sort cuenta con una complejidad temporal que se puede aproximar a O(n \* log2 (n)). Al comparar la taza de crecimiento que tiene el tiempo de ejecución con respecto a la cantidad de Datos en los tres algorítmos se espera que Shell Sort obtenga los resultados más efectivos (en un menor tiempo), puesto que su taza de crecimiento se comporta como una función logarítmica, mientras que en el Selection e Insertion se comporta como una función lineal.

De forma práctica, tanto la Tabla 1 como la Tabla 2 evidencian la superioridad en cuanto a eficacia que posee el algoritmo Shell Sort sobre Selection Sort e Insertion Sort.



Gráfica Shell Sort



Gráfica Insertion Sort y Selection Sort

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

Sí, aunque ambas máquinas cuentan con especificaciones técnicas muy similares, la máquina 2 realizó la mayoría de los procesamientos de forma mucho más efectiva que la máquina 1. No obstante, la máquina 1 logró procesar el ordenamiento de subgrupos aproximadamente 2 veces más grandes que los que logró soportar la máquina 2, puesto que cuando se intentó organizar un conjunto de 64000 datos ocurrió un error de ejecución obteniendo una *blue screen* debido a que se excedió el uso de RAM.

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

Las diferencias se deben principalmente a la forma en la que cada máquina tiene organizada y dividida su memoría RAM. La máquina 1 cuenta con un solo stick de memoría RAM de 16GB, por lo que este puede verse afectado por una mayor cola de procesamiento, mientras que la máquina 2 cuenta con 2 sticks de 8GB de RAM cada uno, permitiendole guardar más información temporal al mismo tiempo. No obstante, estos sticks pueden llenarse más rápido lo que explicaría la aparición de pantallas de error.

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

Si solo tenemos en cuenta los tiempos de ejecución, la mejor estructura de datos que se debe utilizar es la **Array List**, como se puede observar en todas las gráficas, los ordenamientos realizados sobre esta estructura tienen un tiempo de ejecución ligeramente inferior al de su contraparte (Linked List). El motivo de esta diferencia radica en la úbicacion de los datos en memoria puesto en la estructura de tipo Array estos se ubican de forma secuencial, haciendo que sea más fácil que el sistema los encuentre y los organice.