OBSERVACIONES DE LA PRACTICA

Estudiante 1: Valeria Caro Ramírez Cod 202111040

Estudiante 2: Sofia Velasquez Marin Cod 202113334

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Core i5 – 8300H 2.30Ghz | AMD Ryzen 7 3700U 2.30Ghz |
| Memoria RAM (GB) | 16 GB | 16 GB |
| Sistema Operativo | Windows 11 | Windows 10 |

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

# **Maquina 1**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 0.50% | 768 | 244.79 | 26.20 | 20.91 | 20.8 |
| 10.00% | 15008 | 93906.25 | 500 | 1625 | 333.33 |

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (LINKED\_LIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 0.50% | 768 | 16552.09 | 822.92 | 723.95 | 135.42 |
| 10.00% | 15008 | *Más de 20 minutos* | *Más de 20 minutos* | *Más de 20 minutos* | 40031.24 |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* | 93906.25 | *Más de 20 minutos* |
| *Shell Sort* | 500 | *Más de 20 minutos* |
| *Merge Sort* | 1625 | 40031.24 |
| *Quick Sort* | 333.33 | *Más de 20 minutos* |

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

# **Maquina 2**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 0.50% | 768 | 437.5 | 15.625 | 31.25 | 15.625 |
| 100.00% | 15008 | 163046.875 | 828.125 | 2625.0 | 562.5 |

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (LINKED\_LIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 0.50% | 768 | 24531.25 | 1234.375 | 1125.0 | 171.875 |
| 100.00% | 15008 | *Más de 20 minutos* | *Más de 20 minutos* | *Más de 20 minutos* | 59281.25 |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* | 163046.875 | *Más de 20 minutos* |
| *Shell Sort* | 828.125 | *Más de 20 minutos* |
| *Merge Sort* | 625.0 | 59281.25 |
| *Quick Sort* | 2625.0 | *Más de 20 minutos* |

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

# **Preguntas de análisis**

1. ¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?
2. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?
3. De existir diferencias, ¿a qué creen que se deben?
4. ¿Cuál Estructura de Datos funciona mejor si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?
5. Teniendo en cuenta las pruebas de tiempo de ejecución por todos los algoritmos de ordenamiento estudiados (iterativos y recursivos), proponga un ranking de los mismo de mayor eficiencia a menor eficiencia en tiempo para ordenar la mayor cantidad de obras de arte.

**Respuesta a las preguntas de análisis**

1. los algoritmos se comportaron más o menos de la manera esperada por lo visto en la clase, a excepción de Quicksort y Mergesort. Teóricamente Quicksort es más rápido que Mergesort, y a partir de los resultados obtenidos, esto no pasa; puede ser debido a la implementación en el código.
2. Al comparar las tablas de la maquina 1 y 2, se observar que en algunos casos la maquina 1 puede hacer los ordenamientos mas rapidos que la maquina 2. Sin embargo, la diferencia entre los tiempos no es relativamente grande sino por un par de milisegundos.
3. Estas diferencias se pueden dar porque el sistema operativo Windows en algunas ocaciones puede correr aplicaciones en segundo plano como actualizaciones del sistema o del algun programa, que en si ocupan mas memoria y por lo tanto, hacen que otros programas funcionen un poco mas lento. Ademas hay que tener en cuenta que ambos procesadores cuentan con la misma velocidad en GHZ, sin embargo, puede que ambos cuenten con diferentes optimizaciones de desempeño lo que podria dar una variacion en la velocidad del procesador para realizar ciertos procesos.
4. De acuerdo a los datos obtenidos en cada una de las maquinas, podemos concluir que la Estructura de Datos que funciona mejor es la ARRAY\_LIST, debido a que el tiempo de ejecucion al utilizar algoritmos de organizacion con esta estructura es menor y con una gran diferencia de tiempo con la de SINGLE\_LINKED, ya que esta puede llegar a tomar un largo tiempo para que termine de procesar los datos. En las tablas se alcanza a observar que este segundo tipo de estructura puede llegar a tomar mucho mas tiempo en organizar los datos, utilizando algoritmos de sorting como InsertionSort, ShellSort y QuckSort.
5. Ranking
   * + 1. Merge Sort – Mayor Eficiencia
       2. Quick Sort
       3. Shell Sort
       4. Insertion Sort – Menor Eficiencia