OBSERVACIONES DE LA PRACTICA

Estudiante 1: Johan Alejandro Charry - Cod 202111151

Estudiante 2: Nicolas Toro Barrios - Cod 202116790

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel ® Core™ i5-1035G1 CPU | Raizen 5 5500u |
| Memoria RAM (GB) | 8.00 GB | 8.00 GB |
| Sistema Operativo | Windows 10 home 64-bits | Windows 10 home 64-bits |

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]** |
| small | 768 | 2.015,63 | 140,63 | 145,83 | 104,17 |
| 10.00% | 13.815 | No fue posible de calcular | 4.057,29 | 18.000,00 | 2.369,80 |

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]** |
| small | 768 | 16.750,00 | 8.022,92 | 770,84 | 177,09 |
| 10.00% | 13.815 | No fue posible de calcular | No fue posible de calcular | 176.170,00 | 34.453,13 |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* | 2.015,63 | 16.750,00 |
| *Shell Sort* | 140,63 | 8.022,92 |
| *Merge Sort* | 104,17 | 177,09 |
| *Quick Sort* | 145,83 | 770,84 |

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]** |
| small |  |  |  |  |  |
| 10.00% |  |  |  |  |  |

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]** |
| small |  |  |  |  |  |
| 10.00% |  |  |  |  |  |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* |  |  |
| *Shell Sort* |  |  |
| *Merge Sort* |  |  |
| *Quick Sort* |  |  |

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?

**R/** Sí. Teóricamente el algoritmo Insertion Sort es el que peor complejidad tiene y Merge Sort el que mejor complejidad tiene, según las pruebas realizadas se logra comprobar esta efectividad.

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

**R/**

1. De existir diferencias, ¿a qué creen que se deben?

**R/**

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

**R/** La mejor estructura de Datos según los tiempos de ejecución es: **Merge Sort,** ya que tomó la menor cantidad de tiempo en ambas pruebas.

1. 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.

**R/ Top 1:** Merge Sort (Mejor opción en el peor caso y/o mejor caso)

**Top 2:** Shell Sort

**Top 3:** Quick Sort

**Top 4:** Insertion Sort (Peor opción en el peor caso y/0 mejor caso)