OBSERVACIONES DE LA PRACTICA

Santiago Bastos Cod 202110792

Santiago Ramírez Cod 201910908

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz | Ryzen 5 3400G GPU @3.6GHz 4GHz Vega 11 |
| Memoria RAM (GB) | 8,00 GB | 8 GB |
| Sistema Operativo | Windows 10 Home 64-bits | Windows 10 Pro 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]** |
| 0.50% | 147 | 31.30 | 15.60 | 0.0 | 15.6 |
| 100.00% | 294 | 46.90 | 15.60 | 31.25 | 15.6 |

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% | 147 | 109.40 | 31.3 | 31.3 | 31.3 |
| 100.00% | 294 | 859.40 | 78.10 | 78.10 | 31.3 |

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

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

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% | 147 | 15.63 | 0.0 | 0.0 | 0.0 |
| 100.00% | 294 | 57.30 | 10.42 | 0.0 | 10.42 |

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% | 147 | 119.81 | 31.25 | 15.63 | 10.41 |
| 100.00% | 294 | 963.325 | 98.96 | 109.38 | 26.04 |

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

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

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?**

En ARRAY\_LIST el orden de tiempo de ejecución es: Shell sort < Insertion sort < Quick sort < Merge sort. En LINKED-LIST el orden de tiempo de ejecución es: Quick sort < Shell sort < Insertion sort < Merge sort, para muestras inferiores al 5% de los datos. Para muestras superiores al 50% de los datos el orden es: Insertion sort < Shell sort < Merge sort < Quick sort. Teniendo en cuenta todo lo anterior, los datos experimentales no son muy acordes a los teóricos.

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

Si existe una diferencia al ejecutar las pruebas en dos máquinas distintas, sin embargo, estas no fueron muy representativas. Estas diferencias dependen principalmente de la capacidad de computo del procesador de la máquina.

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

Estas diferencias se deben principalmente a la capacidad de analizar datos del procesador, cada uno trabajada a diferente frecuencia y tiene una arquitectura distinta que ayuda al análisis de datos.

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

La estructura de datos que mejor funciona si solo se tiene en cuenta los tiempos de ejecución de los algoritmos es el ARRAYLIST, debido a que su tiempo de ejecución es mucho más bajo que la lista encadenada.

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.**

*Menor eficiencia* < Merge sorts < Quick sorts < Shell sort < Inserion sort < *Mayor eficiencia*