OBSERVACIONES DE LA PRACTICA

Juan Pablo Rodríguez Briceño Cod 202022764

Nicolas Pérez Terán Cod 202116903

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz | M1 |
| Memoria RAM (GB) | 12 GB (9,95 utilizables) | 8 GB |
| Sistema Operativo | Windows 10 Home 64-bits | MacOs Big Sur |

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 | 22890.625 | 1109.375 | 984.375 | 250 |
| 10.00% | 7500 | Indefinido | 38421.875 | Indefinido | 11093.75 |

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 | 21656.25 | 1109.375 | 18546.875 | 171.875 |
| 10.00% | 7500 | Indefinido | 180593.75 | Indefinido | 10921.875 |

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

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

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 | 768 | 10921.45 | 590.77 | 541.17 | 117.95 |
| 10.00% | 7500 | Indefinido | 89908.35 | 140691.98 | 6608.33 |

Tabla 5. 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 | 10835.59 | 586.29 | 528.48 | 117.79 |
| 10.00% | 7500 | Indefinido | 88996.51 | 140088.90 | 6544.04 |

Tabla 6. Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada.

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

Tabla 7. 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/:** Si

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

**R/:** Si

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

**R/:** Esto se debe, posiblemente, a que la maquina 2 sea más rápida con los datos que la maquina 1 o que no tenía tantos procesos necesarios en segundo plano. También puede que haya alguna diferencia en como se tratan los datos y la velocidad de los algoritmos, ya que se trata de una maquina con Windows y otra con MacOS.

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 seria linked\_list ya que, a pesar de que Quick Sort pueda ser más rápida en el Array\_List de la maquina 1, el resto de los algoritmos funcionan mejor con el Array-List tanto para el resto de los algoritmos en la maquina 1 como para todos los algoritmos (incluido Quick Sort) en la maquina 2

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/:** Consideramos que seria así, donde el primero es el mejor y el cuarto es el peor:

1. Merge
2. Shell
3. Quick
4. Insertion