OBSERVACIONES DE LA PRACTICA

Miguel Sandoval Cardozo Cod 201923157

Alejandro Alvarez Jimenez Cod 202020208

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | 1,6 GHz Intel Core i5 de dos núcleos | Intel Core i7-9750H |
| Memoria RAM (GB) | 4GB | 8 GB DDR4 |
| Sistema Operativo | macOS Big Sur 11.5.1 | 8 GB DDR4 |
|  |  | 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]** |
| Small | 768 | 581.51 | 45.32 | 38.43 | 36.72 |
| 10.00% |  | 233245.70 | 1174.13 | 3821.48 | 872.31 |

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 | 28870.30 | 1490.9 | 1362.90 | 190.5 |
| 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* | 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]** |
| 0.50% | 768 | 26015.62 | 1234.375 | 1390.625 | 171.875 |
| 100.00% |  |  |  |  |  |

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]** |
| 0.50% | 768 | 27312.5 | 1937.5 | 1484.375 | 187.5 |
| 100.00% |  |  |  |  |  |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* | 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.

# **Preguntas de análisis**

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

Los algoritmos, se comportan acorde a la teoría, sin embargo, los tiempos de ejecución nos muestran que la eficiencia de estos algoritmos y el tiempo de ejecución dependen de la maquina

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

Efectivamente, podemos notar que, al variar el sistema operativo, ram y procesador se diferencia la eficiencia de un computador sobre el otro a la hora de anotar los tiempos de ejecución.

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

Primero, al sistema operativo, seguramente Windows almacena de manera distinta los datos a lo que hace iOS, también el procesador juega un papel fundamental, pues entre mejor sea el procesador mas rápido se van a mover los datos, y finalmente la RAM dicta la memoria disponible, si la RAM es pequeña seguramente la mayoría de la memoria disponible va a estar ocupándose de que al computador no le entren virus y que funcione correctamente, dejando poco espacio para que el programa se ejecute con la mayor utilizando el 100% del procesador.

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

Arraylist

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.

1.Merge

2.Quick

3.ShellSort

4.SelectSort

5.Insert