OBSERVACIONES DE LA PRACTICA

Camilo Ortiz Cruz (1) Cod 201821615

Kevin Fernando Gómez Camargo (2) Cod 202015120

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz | 2,5 GHz Intel Core i5 de dos núcleos |
| Memoria RAM (GB) | 16 | 4 |
| Sistema Operativo | Windows 10 Home 64-bits | macOS Catalina Versión 10.15.7 |

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]** |
| 0.5% | 768 | 165.62 | 6.25 | 12.496 | 9.372 |
| 10% | 15008 | 71859.37 | 353.13 | 1112.5 | 243.75 |
| 20% | 29489 | X | 768.75 | 3925 | 490.62 |
| 100.00% | 138150 | X | 4674.99 | X | 2678.12 |

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]** |
| 0.5% | 768 | 10821.876 | 500 | 462.49 | 56.25 |
| 10% | 15008 | X | X | X | 27900.0 |
| 100.00% | 138150 | X | X | X | X |

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 | 608.49 | 39.18 | 35.31 | 34.13 |
| 10.00% | 15008 | 237531.43 | 1219.63 | 3957.32 | 855.11 |

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 | 31489.98 | 1594.62 | 1389.42 | 201.24 |
| 10.00% | 15008 | X | X | X | 81158.97 |

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 – Maquina 2**

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

No, aunque los algoritmos de ordenamiento recursivos Quick Sort y Merge y el ordenamiento iterativo Shell Sort fueron buenas opciones en general, Quick Sort no fue más rápido que Merge en ningún caso.

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

Sí, el tiempo depende bastante de la maquina en que se ejecuten los cambios.

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

A la velocidad y capacidad de procesamiento de datos, a la disponibilidad de memoria RAM para almacenar las instrucción y datos que sean necesarios a la hora de correr los algoritmos de ordenamiento y al sistema operativo ya que algunos pueden administrar mejor los recursos del computador.

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 funciona mejor según los tiempo de ejecución de todas las pruebas es el arreglo (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 Sort
       2. Shell Sort
       3. Quick Sort
       4. Insertion Sort