OBSERVACIONES DE LA PRACTICA

Daniel Alonso Cod 201419873

Nicolas Díaz Montaña Cod 202021006

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel® Core™ i5- 8250U CPU @ 1,60 GHz, 1.8 Ghz | 2,7 GHz Intel Core i5 de cuatro núcleos |
| Memoria RAM (GB) | 8GB | 16 GB 1600 MHz DDR3 |
| Sistema Operativo | Windowa 10 Home 64-bits | macOS Catalina Version. 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]** |
| 0.50% | 768 | 18901.04 | 1072.91 | 947.91 | 226.04 |
| 100.00% | 1008 | 46843.75 | 1750 | 1828.12 | 250 |

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 | 18900.01 | 1070.04 | 947.70 | 226.21 |
| 100.00% | 1008 | 46801.82 | 1750 | 1827.9 | 250 |

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 | 24960.51 | 1310.85 | 1222.82 | 198.95 |
| 10.00% | 1008 | 58901.70 | 2354.68 | 2405.55 | 329.45 |

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 | 25929.51 | 1374.47 | 1221.79 | 202.45 |
| 10.00% | 1008 | 58348.69 | 2324.29 | 2385.13 | 330.52 |

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?

Si es correcto, se puede evidenciar que insertion sort es el más lento de los algoritmos mientras que merge sort es el más rapido, estando el medio tanto Shell sort y quick sort dependiendo de la situación.

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

Si

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

Se pudo evidenciar que en la maquina dos hay más tiempo de carga, lo más probable es que se deba al procesador. Se llego a la conclusión es por que aunque la Maquina 2 tenga más RAM y tenga un sistema operativo equivalente al de la maquina 1, este sigue siendo más lento.

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

Según los datos que evidenciamos la mejor estructura de datos resulta siendo la lista enlazada a la hora de medir el timepo de ejecución de los algoritmos, aún así la diferencia no es tan significativa.

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.

El primero debe ser Merge Sort debido a como por su metodo de división de los elementos hace que su tiempo de ejecución sea por mucho el mejor. El segundo seria Quick Sort, aunque siendo parecido al merge su limitante llega el pivote que almacena. Como medalla de bronze llega el Shell Sort, no siendo el peor en terminos de ejecución pero no a la misma velocidad que los otros dos sort anteriores. En el ultimo lugar esta el insertion sort, teniendo la carga más lenta con diferencia debido a que debe hacer varias itertaciones para poder ordenar sus elementos.