OBSERVACIONES DE LA PRACTICA

Cristian Armando Sánchez Ocampo 202022112

Luis Felipe Dussán R- 201912309

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD Ryzen 3 2200G 3.85 GHz | 1,4 GHz Quad-Core Intel Core i5 |
| Memoria RAM (GB) | 8.00GB | 8 GB 2133 MHz LPDDR3 |
| Sistema Operativo | Windows 10Pro 64 bits | Mac.OS Big Sur. Versión 11.2.1 |

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

# **Maquina 1**

Límite de carga de datos: 10-12 min.

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 687.5 | 812.5 | 31.25 | 23.43 | 31.25 |
| 2000 | 2073.12 | 3195.5 | 93.75 | 62.5 | 67.7 |
| 4000 | 11000 | 13507.81 | 203.12 | 132.81 | 140.62 |
| 8000 | 43812.5 | 54515.62 | 492.19 | 335.93 | 281.25 |
| 16000 | 187093.75 | 224328.12 | 1062.5 | 598.25 | 601.56 |
| 32000 | 755578.12 | Pasa el rango | 2546.87 | 1286.46 | 1312.5 |
| 64000 | Pasa el rango | Pasa el rango | 6156.25 | 2687.5 | 2804.69 |
| 128000 | Pasa el rango | Pasa el rango | 14687.5 | 5859.37 | 6015.12 |
| 256000 | Pasa el rango | Pasa el rango | 37812.5 | 12921.87 | 12871.25 |
| 375942 | Pasa el rango | Pasa el rango | 59703.12 | 28218.75 | 19468.75 |

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (LINKED\_LIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 44110.45 | 39748.5 | 2203.12 | 1679.69 | 210.93 |
| 2000 | 383531.25 | 333609.38 | 10890.88 | 8593.75 | 859.37 |
| 4000 | Pasa el rango | Pasa el rango | 51953.12 | 37328.12 | 3515.62 |
| 8000 | Pasa el rango | Pasa el rango | 263218.75 | 198015.62 | 14546.87 |
| 16000 | Pasa el rango | Pasa el rango | Pasa el rango | 722171.88 | 61609.38 |
| 32000 | Pasa el rango | Pasa el rango | Pasa el rango | Pasa el rango | 261343.75 |
| 64000 | Pasa el rango | Pasa el rango | Pasa el rango | Pasa el rango | ‎ Pasa el rango ‏‏ |
| 128000 | Pasa el rango | Pasa el rango | Pasa el rango | Pasa el rango | Pasa el rango |
| 256000 | Pasa el rango | Pasa el rango | Pasa el rango | Pasa el rango | Pasa el rango |
| 512000 | Pasa el rango | Pasa el rango | Pasa el rango | Pasa el rango | Pasa el rango |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | Mejor (promedio suma de tiempos obtenidos =  4359,469) | Mejor (promedio suma de tiempos obtenidos =  57014,32) |
| Quick sort | Peor (promedio suma de tiempos obtenidos =  5212,687 | Peor (promedio suma de tiempos obtenidos =  193557,812) |

Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
  + Comparación de rendimiento ARRAYLIST.
  + Comparación de rendimiento LINKED\_LIST.
  + Comparación de rendimiento para Insertion Sort.
  + Comparación de rendimiento para Selection Sort.
  + Comparación de rendimiento para Shell Sort.
  + Comparación de rendimiento para MergeSort.
  + Comparación de rendimiento para QuickSort.

# **Maquina 2**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | **624,64** | **816,35** | **48,57** | **38,66** | **41,21** |
| 2000 | **2482,18** | **3235,76** | **96,26** | **72,60** | **74,81** |
| 4000 | **10066,48** | **12847,42** | **195,66** | **142,23** | **138,54** |
| 8000 | **40303,85** | **55838,42** | **471,83** | **325,44** | **277,65** |
| 16000 | **168056,40** | **210807,52** | **1013,28** | **592,8** | **585,59** |
| 32000 | **695717,31** | **856012,88** | **2387,9** | **1250,58** | **1249,39** |
| 64000 | **Superó límite** | **Superó límite** | **5590,71** | **2553,08** | **2654,72** |
| 128000 |  |  | **13377,14** | **5590,25** | **5663,06** |
| 256000 |  |  | **34429,71** | **12297,31** | **12027,93** |
| **375942** |  |  | **56059,12** | **27108,51** | **18141,24** |

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (LINKED\_LIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | **41691,13** | **37361,68** | **2101,29** | **1567,11** | **211,75** |
| 2000 | **344157,43** | **307570,53** | **9912,26** | **7786,07** | **817,03** |
| 4000 | **Superó límite** | **Superó límite** | **45757,21** | **32786,85** | **3280,35** |
| 8000 |  |  | **230355,33** | **172197,46** | **12821,49** |
| 16000 |  |  | **Superó límite** | **620970,94** | **51591,28** |
| 32000 |  |  |  | **Superó** | **215541,98** |
| 64000 |  |  |  |  | **Superó límite** |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | Más eficiente | Segundo menos eficiente |
| Quick sort | Segundo más eficiente | Menos eficiente |

Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  + Comparación de rendimiento ARRAYLIST.
  + Comparación de rendimiento LINKED\_LIST.
  + Comparación de rendimiento para Insertion Sort.
  + Comparación de rendimiento para Selection Sort.
  + Comparación de rendimiento para Shell Sort.
  + Comparación de rendimiento para MergeSort.
  + Comparación de rendimiento para QuickSort.

# **Preguntas de análisis**

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

* Si, con esta mejora en la implementación del código, aparte de poder correr una buena cantidad de datos en poco tiempo, las funciones de ordenamiento se comportan como se esperaría teóricamente como las funciones más rápidas quick y merge y la más lenta selection.

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

* Si, hay una pequeña diferencia en el array\_list siendo la maquina 1 más rápida solamente con una cantidad de datos pequeñas. Exceptuando este caso, la maquina 2 es un poco más rápida.

1. De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

* Creemos que esta diferencia se debe a la velocidad y capacidad de la memoria RAM que tiene la maquina 2 y no afecta tanto el procesador cuando hablamos de una cantidad grande de datos.

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

* Merge es un buen candidato, puesto que, aunque al inicio en un array\_list se demoraba un poco más que la función quick, al final, con una mayor cantidad de datos, esta dio mejores resultados. Además, en un linked\_list, nos permitió trabajar con una cantidad de datos mayor a las demás funciones. Un ejemplo la eficiencia en 32000 datos con 1/3 del tiempo, aproximadamente, que la función quick con la mitad de estos datos.

1. Para el caso analizado de ordenamiento de los videos, teniendo en cuenta los resultados de tiempo reportados por todos los algoritmos de ordenamiento estudiados (iterativos y recursivos), proponga un ranking de los algoritmos de ordenamiento (de mayor eficiencia a menor eficiencia en tiempo) para ordenar la mayor cantidad de videos.

|  |  |
| --- | --- |
| N° | Algoritmo de ordenamiento |
| 1 | Merge |
| 2 | Quick |
| 3 | Shell |
| 4 | Insertion |
| 5 | Selection |