OBSERVACIONES DE LA PRACTICA

Ivan Camilo Ballén Méndez Cod 202011440

María José Sáenz Rodríguez Cod 202013542

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD R5 2.1 GHz - 3.7 GHz | Intel®️ Core™️ i5-8300H |
| Memoria RAM (GB) | 8.0 GB | 8.0 GB |
| Sistema Operativo | Windows 10 Pro-64-bits | Windows 10 - 64 bits |

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

# **Maquina 1**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 65010.42 | 64640.63 | 3432.29 | 3135.42 | 411.46 |
| 2000 | 523270.83 | 355828.13 | 19776.04 | 14062.50 | 1562.50 |
| 4000 | Tiempo Excedido | Tiempo Excedido | 88427.21 | 38958.33 | 5244.79 |
| 8000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | 170427.08 | 15718.75 |
| 16000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | 734843.75 | 64161.46 |
| 32000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | 393666.67 |
| 64000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido |
| 128000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido |
| 187971 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido |
|  |  |  |  |  |  |

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 | 47135.42 | 43338.54 | 2244.79 | 3187.50 | 401.04 |
| 2000 | 382750.00 | 352713.54 | 12067.71 | 14177.08 | 1578.13 |
| 4000 | Tiempo Excedido | Tiempo Excedido | 56937.50 | 60526.04 | 6031.25 |
| 8000 | Tiempo Excedido | Tiempo Excedido | 255765.63 | 260895.83 | 23604.17 |
| 16000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | 96369.79 |
| 32000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | 396713.54 |
| 64000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido |
| 128000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido |
| 187971 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido |
|  |  |  |  |  |  |

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 | X |  |
| Quick sort | X |  |

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 | 37239,58 | 36369,79 | 1723,26 | 3380.21 | 359.38 |
| 2000 | 355234,37 | 305083,33 | 9932,29 | 17302.08 | 1744.79 |
| 4000 | 2980838,54 | 2281598,96 | 44130,21 | 74567.71 | 7307.29 |
| 8000 | Tiempo Excedido | Tiempo Excedido | 206234,37 | 160098.96 | 20046.86 |
| 16000 | Tiempo Excedido | Tiempo Excedido | 1012786,40 | 683223.96 | 60354.17 |
| 32000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | 3387281.25 | 260567.71 |
| 64000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | 1016906.25 |
| 128000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido |
| 187971 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido |
|  |  |  |  |  |  |

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 | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | 1781.25 | 239.58 |
| 2000 | 41369,79 | 39562,50 | 2447,92 | 8286.46 | 937.5 |
| 4000 | 330947,92 | 302375,00 | 10656,25 | 33609.38 | 3052.08 |
| 8000 | Tiempo Excedido | Tiempo Excedido | 53171,87 | 145286.46 | 12635.42 |
| 16000 | Tiempo Excedido | Tiempo Excedido | 221385,42 | 602442.71 | 50734.38 |
| 32000 | Tiempo Excedido | Tiempo Excedido | 1125286,40 | 3145203.13 | 241760.42 |
| 64000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | 808916.67 |
| 128000 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido |
| 187971 | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido | Tiempo Excedido |
|  |  |  |  |  |  |

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 |  | x |
| Quick sort |  | x |

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?

En los experimentos se puede apreciar claramente la predicción teórica que se realizó. El algoritmo Quick Sort resulta ser menos eficiente que el algoritmo Merge Sort en cuanto si se da un caso cercano a su peor caso tiende a tener una complejidad temporal que crece con el cuadrado de la distancia. Por otro lado, el algoritmo Merge mantiene una tendencia linearítmica y de hecho podemos concluir que es el algoritmo más eficiente si tomamos el límite cuando la cantidad de datos tiende a infinito. También observamos que la predicción teórica en cuanto la complejidad temporal de Shell y Quick tienden a ser muy parecidas, lo que coincide con la predicción (en el mejor caso ambos se comportan de forma linearítmica y en el peor la diferencia es de un 0.5 en el exponente)

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

Sí. Una de las diferencias evidenciadas en las tablas y gráficas corresponde a los tiempos de ejecución, puesto que en la maquina 1 el procesamiento de los datos es más tardío que en la maquina 2, independientemente del tipo de ordenamiento en el que se esté trabajando. Además, teniendo en cuenta el tiempo límite establecido (15 minutos), se determina que las maquinas no logran procesar la misma cantidad de datos, por lo que se obtiene una cifra diferente de resultados reflejado en las tablas.

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

A pesar de las semejanzas en las tendencias de los gráficos en cada máquina, la diferencia más grande y la que se le debería encontrar una explicación se corresponde a la elección del algoritmo más eficiente para cada estructura de datos. Mientras que para la primera maquina se encontraron resultados favorables con Array List, la segunda maquina muestra un mejor rendimiento con el Linked List. Debido a que en ambas maquinas se usó el mismo código iterativo para automatizar y las pruebas y el mismo interprete, se conjetura que la raíz de esta diferencia debe de estar relacionada con las diferencias en los procesadores. La hipótesis es que, si replicamos el experimento, pero esta vez con los mismos procesadores, los resultados se inclinaran únicamente hacia un tipo de estructura de datos para cada algoritmo. También debemos considerar dentro de las posibles causas de esta diferencia los factores de error experimentales. Con esto se refiere al caso en que durante los experimentos la computadora estuviera atendiendo tareas que no se pueden finalizar por el administrador de tareas.

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

De acuerdo con los resultados obtenidos, se establece una contradicción en la definición de cual estructura de datos es más eficiente para las dos máquinas. Sin embargo, teniendo en cuenta que la maquina 2 presenta los tiempos de ejecución más pequeños, se determina que la estructura de datos más eficiente es la representación de lista enlazada (LINKED\_LIST).

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.

Según los tiempos de ejecución reportados en los diferentes ordenamientos se realiza el siguiente ranking:

1. Merge Sort

2. Quick Sort

3. Shell Sort

4. Selection Sort

5. Insertion Sort