OBSERVACIONES DE LA PRACTICA

Estudiante 1 Felipe Rueda Cod 202010903

Estudiante 2 Julian Parra Cod 202013033

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel® Core™ i5-1035G7 CPU @1.20GHz 1.50GHz | Intel® Core™ i7-10870H CPU @2.20GHz 2.21GHz |
| Memoria RAM (GB) | 8.00 GB | 8.00 GB |
| Sistema Operativo | Windows 10 pro-64 bits | Windows 10 pro-64 bits |

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

# **Maquina 1**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) | Quick Sort (ms) | Merge sort (ms) |
| 1000 | 1276.42 | 855.417 | 479.16 | 1109.375 | 41.67 |
| 2000 | 5187.5 | 3578.125 | 2041.67 | 2734.375 | 52 |
| 4000 | 28317.04 | 26338.54 | 10213.54 | 10395.83 | 93.75 |
| 8000 | 89223.95 | 65656.25 | 66708.33 | Tiempo excedido | 213.54 |
| 16000 | 531791.33 | 285140.625 | 159125.0 | Tiempo excedido | 395.83 |
| 32000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 760.4 |
| 64000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 1583.33 |
| 128000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 256000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 512000 | 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 | 69760.22 | 50255 | 34791.76 | 37382.81 | 145.83 |
| 2000 | 613265.625 | 404609.375 | 423031.25 | 294432.29 | 552.08 |
| 4000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 2067.7 |
| 8000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 8864.58 |
| 16000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 35238.28 |
| 32000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 147203.125 |
| 64000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 128000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 256000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 512000 | 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 (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) | Quick Sort (ms) | Merge Sort (ms) |
| 1000 | 1151.042 | 765.625 | 406.25 | 614.58 | 26.04 |
| 2000 | 4063.708 | 2916.7 | 1661.458 | 2473.96 | 52.08 |
| 4000 | 16557.292 | 12927.083 | 6776.041 | 8510.42 | 93.75 |
| 8000 | 71328.125 | 51015.625 | 26296.875 | Tiempo excedido | 143.83 |
| 16000 | 314250 | 252489.583 | 129203.125 | Tiempo excedido | 296.88 |
| 32000 | 1437390.625 | 876390.625 | 544963.542 | Tiempo excedido | 635.42 |
| 64000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 1416.67 |
| 128000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 2880.21 |
| 256000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 6041.67 |
| 512000 | 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 | 63385.417 | 136578.125 | 32000 | 32604.17 | 135.42 |
| 2000 | 479723.958 | 381666.667 | 264369.792 | 258927.08 | 520.83 |
| 4000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 2010.42 |
| 8000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 8359.38 |
| 16000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 32838.54 |
| 32000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 131385.42 |
| 64000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 530718.75 |
| 128000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 256000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 512000 | 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) |
| Insertion sort | X |  |
| Selection 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?

El comportamiento de los algoritmos fue acorde a los enunciados teóricamente porque Merge Sort es el algoritmo más rápido en organizar, pero gasta mucho espacio temporal, además funciona bien para listas encadenadas y siempre mantiene los mismos casos de complejidad. Además, se evidencio que Quick sort es inestable en su rendimiento, como se observo en los resultados obtenidos.

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

Si existe una diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes maquinas pues se evidencio que la maquina con mejor procesador tuvo menor tiempo en ejecutar los algoritmos de ordenamiento, en cambio la maquina con un procesador inferior a la otra maquina tuvo un tiempo mayor en ejecutar los algoritmos.

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

Se cree que estas diferencias se deben a que las maquinas tienen diferentes procesadores, por tanto cada procesador hace las operaciones de los algoritmos más rápido.

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

Se puede evidenciar que la mejor estructura de datos para utilizar es el tipo arreglo Array List pues se puede evidenciar que con la ejecución de los algoritmos tanto iterativos como recursivos. Esto significa que en comparación al Linked List los tiempos de ejecución son muy lentos o tiene mayor tiempo para responder. Además de que con los tamaños de muestra más grandes que 4000 el tiempo es exagerado excepto en Merge sort (en Merge sort empieza a tener tiempo excedido cuando el tamaño de la muestra es de 64000) lo que no pasa en Array 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.

Con base a los resultados obtenidos por las dos maquinas se puede realizar el siguiente ranking, aunque en terminos generales el más rápidos o con mayor eficiencia es el Merge Sort y el que tiene menor eficiencia es el insertion Sort.

1. Merge Sort (recursivo)
2. Shell Sort (iterativo)
3. Quick Sort (recursivo)
4. Selection Sort (iterativo)
5. Insertion Sort (iterarivo)