OBSERVACIONES DE LA PRACTICA

Juan Camilo Falla 201922219

Nicolás Klopstock 202021352

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 (M.1) | Máquina 2 (M.2) |
| Procesadores | i7 10700 @2.9GHz- 4.8GHz | i5 6200U CPU @2.30GHz 2.40GHz |
| Memoria RAM (GB) | 16 gb | 4 gb |
| Sistema Operativo | Windows 10 Home | Windows 10 Home |

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

# **Maquina 1**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra** | **Insertion Sort** | **Selection Sort** | **Shell Sort** | **Quick Sort** | **Merge Sort** |
| **(ARRAY\_LIST)** | **(ms)** | **(ms)** | **(ms)** | **(ms)** | **(ms)** |
| **1000** | 343.75 | 359.38 | 15.63 | 15.63 | 15.63 |
| **2000** | 1437.5 | 1515.25 | 31.25 | 31.25 | 31.25 |
| **4000** | 5890.63 | 6187.5 | 93.75 | 62.5 | 62.5 |
| **8000** | 23531.25 | 24406.25 | 234.38 | 140.63 | 140.63 |
| **16000** | 94984.38 | 102109.38 | 546.88 | 312.5 | 312.5 |
| **32000** | 390687.5 | 429140.63 | 1343.75 | 656.25 | 671.88 |
| **64000** | Excede 12 mins | Excede 12 mins | 3125 | 1515.625 | 1468.75 |
| **128000** | Excede 12 mins | Excede 12 mins | 7781.25 | 3062.5 | 3125 |
| **256000** | Excede 12 mins | Excede 12 mins | 18406.25 | 6828.13 | 6750 |
| **512000** | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. |

*Tabla 1. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra** | **Insertion Sort** | **Selection Sort** | **Shell Sort** | **Quick Sort** | **Merge Sort** |
| **(LINKED\_LIST)** | **(ms)** | **(ms)** | **(ms)** | **(ms)** | **(ms)** |
| **1000** | 26750 | 24546.88 | 1234.38 | 1000 | 140.63 |
| **2000** | 230921.88 | 205937.5 | 6921.88 | 5078.13 | 531.25 |
| **4000** | Excede 12 mins | Excede 12 mins | 33296.88 | 23515.63 | 2171.88 |
| **8000** | Excede 12 mins | Excede 12 mins | 146843.75 | 106484.38 | 8625 |
| **16000** | Excede 12 mins | Excede 12 mins | 698015.63 | 377859.38 | 34948.38 |
| **32000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins | 140203.13 |
| **64000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins | 570687.5 |
| **128000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins |
| **256000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins |
| **512000** | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Quick sort | X |  |
| Merge 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** | **Insertion Sort** | **Selection Sort** | **Shell Sort** | **Quick Sort** | **Merge Sort** |
| **(ARRAY\_LIST)** | **(ms)** | **(ms)** | **(ms)** | **(ms)** | **(ms)** |
| **1000** | 703,13 | 781,25 | 46,875 | 31,25 | 46,83 |
| **2000** | 3171,88 | 3656,25 | 93,75 | 62,5 | 93,75 |
| **4000** | 12750 | 12843,75 | 250 | 156,25 | 125 |
| **8000** | 52203,13 | 53015,63 | 531,25 | 265,63 | 296,83 |
| **16000** | 211265,63 | 221625 | 1218,75 | 625 | 593,75 |
| **32000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | 1187,5 | 1343,75 |
| **64000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | 2765,63 | 2656,25 |
| **128000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | 5609,38 | 5875 |
| **256000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | 12625 | 1203,13 |
| **512000** | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra** | **Insertion Sort** | **Selection Sort** | **Shell Sort** | **Quick Sort** | **Merge Sort** |
| **(LINKED\_LIST)** | **(ms)** | **(ms)** | **(ms)** | **(ms)** | **(ms)** |
| **1000** | 68890,63 | 69296,88 | 3109,38 | 2437,50 | 265, 63 |
| **2000** | Excede 12 mins | 499843,75 | 16281,25 | 9031,25 | 1062,5 |
| **4000** | Excede 12 mins | Excede 12 mins | 82,67 | 45921,88 | 4125,00 |
| **8000** | Excede 12 mins | Excede 12 mins | 360,49 | 67968,74 | 17109,38 |
| **16000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | 82967,50 | 70312,50 |
| **32000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins | 285421.88 |
| **64000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins |
| **128000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins |
| **256000** | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins | Excede 12 mins |
| **512000** | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. | Excede Tamano archivo. |

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Quick sort | X |  |
| Merge sort | X |  |

**Graficas**

* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la Maquina 2.
* Comparación de rendimiento ARRAYLIST.
* Comparación de rendimiento LINKEDLIST.
* Comparación de rendimiento 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 sí esta de acorde a lo que se plantea teóricamente al momento de realizar las pruebas con arreglos, los ordenes de crecimiento son bastante precisos y las pequeñas variaciones pueden surgir a causa de la máquina utilizada para realizar las pruebas. Sin embargo, nuevamente se observa que los ordenes de crecimiento de la ejecución con listas enlazadas no va de acuerdo a la teoría. Se ve un orden de crecimiento como potencia pero estos suelen ser mayores a lo esperado. La demora adicional se debe a que el programa debe redefinir los apuntadores y los nodos cada vez que se realiza el ordenamiento.

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

Si exísten diferencias y, en casi todos las evaluaciones, son altas. En las funciónes más rápidas (Quick, Merge y Shell en ARRAY\_LIST), los tiempos con pocos datos son parecidos. Pero en LINKED\_LIST, con cualquier método de ordenamiento las diferencias son muy grandes.

3) De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

Suponemos que es por temas de procesador, las dos máquinas tienen diferentes tipos de procesador y memoria que suponemos que afecta en gran parte a los tiempos de ordenamiento y carga.

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

La mejor estructura de datos para realizar los ordenamientos es el arreglo, ya que este muestra un tiempo de ejecución significativamente menor que cuando se hacen los ordenamientos de la lista enlazada.

5) 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 nuestros resultados, el ranking sería:

1. Merge
2. Quick
3. Shell
4. Selection
5. Insertion