OBSERVACIONES DE LA PRACTICA

Esteban Gonzales 202021225

Abel Arismendy 202020625

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD Ryzen 5 |  |
| Memoria RAM (GB) | 8 |  |
| Sistema Operativo | Windows 10 |  |

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 | 890.625 | 812.5 | 62.5 | 46.875 | 62.5 |
| 2000 | 3531.25 | 3921.875 | 171.875 | 93.75 | 93.75 |
| 4000 | 16859.375 | 15781.25 | 328.125 | 234.375 | 234.375 |
| 8000 | 70000.0 | 67156.25 | 750.0 | 359.375 | 484.375 |
| 16000 | 296640.625 | 285031.25 | 1437.5 | 750.0 | 812.5 |
| 32000 | 15+ min | 15+ min | 3578.125 | 1937.5 | 1843.75 |
| 64000 |  |  | 9000.0 | 3750.0 | 3640.625 |
| 128000 |  |  | 19984.375 | 8187.5 | 8734.375 |
| 256000 |  |  | 49656.25 | 16671.875 | 18609.375 |
| 512000 |  |  | No suficientes elementos | No suficientes elementos | No suficientes elementos |

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 | 59656.25 | 69750.0 | 3093.75 | 2921.875 | 328.125 |
| 2000 | 508921.875 | 608703.125 | 15765.625 | 11031.25 | 1140.625 |
| 4000 | 15+ min | 15+ min | 77484.375 | 54171.875 | 5687.5 |
| 8000 |  |  | 397609.375 | 223031.25 | 19906.25 |
| 16000 |  |  | 15+ min | 15+ min | 85578.125 |
| 32000 |  |  |  |  | 15+ min |
| 64000 |  |  |  |  |  |
| 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 |  | Mejor |
| Quick sort | Mejor |  |

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.

Chart, line chart

Description automatically generated

* + Comparación de rendimiento LINKED\_LIST.

Chart, line chart

Description automatically generated

* + Comparación de rendimiento para Insertion Sort.

Chart

Description automatically generated

* + Comparación de rendimiento para Selection Sort.

Chart, line chart

Description automatically generated

* + Comparación de rendimiento para Shell Sort.

A picture containing chart

Description automatically generated

* + Comparación de rendimiento para MergeSort.

Chart, bar chart, histogram

Description automatically generated

* + Comparación de rendimiento para QuickSort.

Chart, line chart

Description automatically generated

# **Maquina 2**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 |  |  |  |  |  |
| 2000 |  |  |  |  |  |
| 4000 |  |  |  |  |  |
| 8000 |  |  |  |  |  |
| 16000 |  |  |  |  |  |
| 32000 |  |  |  |  |  |
| 64000 |  |  |  |  |  |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

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 |  |  |  |  |  |
| 2000 |  |  |  |  |  |
| 4000 |  |  |  |  |  |
| 8000 |  |  |  |  |  |
| 16000 |  |  |  |  |  |
| 32000 |  |  |  |  |  |
| 64000 |  |  |  |  |  |
| 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 |  |  |
| Quick sort |  |  |

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 es acorde a lo enunciado teóricamente.

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

Si existen diferencias entre los resultados obtenidos al ejecutar las pruebas en diferentes maquinas.

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

Nosotros creemos que las diferencias se deben a las diferencias en hardware como el procesador y la cantidad de RAM.

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

El Array List es mucho mejor ya que en todo logro obtener mucho mejores tiempos de ejecución.

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.
   * + Quick Sort
     + Merge Sort
     + Shell Sort
     + Insertion Sort
     + Selection Sort