# OBSERVACIONES DE LA PRACTICA

Estudiante 1 Cod XXXX

Estudiante 2 Cod XXXX

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R)Core(TM) i7-8750H CPU @ 2.20GHz 2.20 GHz | Intel(R)Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz |
| Memoria RAM (GB) | 16.0 GB 2666 Mhz | 16.0 GB 2666 Mhz |
| Sistema Operativo | Windows 10 Home - 64bits | Windows 10 Home - 64bits |

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 | 562,5 | 625 | 46,875 | 15.625 | 31.25 |
| 2000 | 2343,75 | 2609,375 | 78,125 | 62.5 | 62.5 |
| 4000 | 8875 | 10656,25 | 156,25 | 156.25 | 109.375 |
| 8000 | 37125 | 43203,125 | 390,625 | 250.0 | 234.375 |
| 16000 | 146906,5 | 182875 | 968,75 | 531.25 | 484.375 |
| 32000 | 620171,875 | 756250 | 2109,375 | 1171.875 | 1093.75 |
| 64000 |  |  | 5125 | 2515.625 | 2296.875 |
| 128000 |  |  | 12218,75 | 5000.0 | 5062.5 |
| 256000 |  |  | 30062 | 12234.375 | 10843.75 |
| 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 | 38234,375 | 37546,865 | 2171,875 | 1468.75 | 218.75 |
| 2000 | 326781,25 | 274531 | 8906,25 | 7281.25 | 921.875 |
| 4000 | 2668125 | 2288343 | 39078,125 | 30750.0 | 3718.75 |
| 8000 |  |  | 204359,375 | 135703.125 | 12484.375 |
| 16000 |  |  | 1002468,75 | 541468.75 | 58468.75 |
| 32000 |  |  |  |  | 224625.0 |
| 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 | x | x |
| 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 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 | 484.375 | 578.125 | 31.25 | 15.625 | 15.625 |
| 2000 | 1984.375 | 2437.5 | 62.5 | 46.875 | 46.875 |
| 4000 | 8000 | 9640.625 | 140.625 | 109.375 | 93.75 |
| 8000 | 33437.5 | 39875 | 343.75 | 218.75 | 203.125 |
| 16000 | 130531.25 | 159953.125 | 812.5 | 437.5 | 437.5 |
| 32000 | 552078.125 | 688796.875 | 1796.875 | 968.75 | 906.25 |
| 64000 |  |  | 4390.625 | 2140.625 | 1984.375 |
| 128000 |  |  | 10718.75 | 4421.875 | 4250 |
| 256000 |  |  | 26843.75 | 10187.5 | 9156.25 |
| 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 | 31531.25 | 27687.5 | 1656.25 | 1093.75 | 171.875 |
| 2000 | 257203.125 | 225812.5 | 6765.625 | 5859.375 | 703.125 |
| 4000 |  |  | 34218.75 | 26015.625 | 2765.625 |
| 8000 |  |  | 171390.625 | 102718.75 | 10937.5 |
| 16000 |  |  | 791140.625 | 431937.5 | 43875 |
| 32000 |  |  |  |  | 179343.75 |
| 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 | x | x |
| 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?
2. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?
3. De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?
4. ¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?
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.

**SOLUCION**

1. Si van acorde a la teoría y son más rápidos que el shellsort
2. Si, en la máquina 2 el ordenamiento se realiza ligeramente más rápido.
3. A la fecha de lanzamiento de los procesadores. Por lo que la arquitectura de estos es diferente y el de la máquina dos es más reciente.
4. El Merge Sort.

* Merge Sort
* Quick Sort
* Shell Sort
* Selection Sort / Insertion Sort