OBSERVACIONES DE LA PRACTICA

Juan Camilo Colmenares Ortiz Código 202011866

Juan Andrés Ospina Sabogal Código 202021727

# Máquina 1 Máquina 2

|  |  |  |
| --- | --- | --- |
| **Procesadores** | AMD Ryzen 7 4800H 2.90 GHz | Intel® Core ™ i5-7267U @3.1GHz |
| **Memoria RAM (GB)** | 16 Gb | 8.0 Gb |
| **Sistema Operativo** | Windows 10 | macOS 11.2.1 Big Sur |

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

# Maquina 1

## Resultados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **T**amaño de la muestra (ARRAYLIST) | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort** | **Merge Sort** |
| 1000 | 406.25 | 484.375 | 15.625 | 15.630 | 15.630 |
| 2000 | 1687.50 | 1937.50 | 62.50 | 31.25 | 36.460 |
| 4000 | 6968.75 | 8171.875 | 125.00 | 93.75 | 83.33 |
| 8000 | 28734.375 | 37953.125 | 312.50 | 203.130 | 177.08 |
| 16000 | 124296.875 | 152125.0 | 671.88 | 359.38 | 385.42 |
| 32000 | 506296.875 | 612468.75 | 1593.75 | 833.33 | 817.71 |
| 64000 | 2408296.875 | 2437640.625 | 3765.63 | 1817.71 | 1755.21 |
| 128000 |  |  | 8375.00 | 3916.67 | 3781.25 |
| 256000 |  |  | 23265.63 | 9281.25 | 8098.96 |
| 375942 |  |  | 37562.50 | 18473.958 | 12244.79 |

*Tabla 2. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **T**amaño de la muestra (LINKED\_LIST) | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort** | **Merge Sort** |
| 1000 | 29859.375 | 24562.50 | 1406.25 | 1739.58 | 140.63 |
| 2000 | 238046.875 | 200703.13 | 6828.125 | 8937.50 | 609.38 |
| 4000 | 1957875.00 | 1640578.13 | 30703.13 | 36239.58 | 2145.83 |
| 8000 |  |  | 154937.50 |  | 8677.08 |
| 16000 |  |  | 733906.25 |  | 37026.04 |
| 32000 |  |  | 3719875.00 |  | 158453.13 |
| 64000 |  |  |  |  |  |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 375942 |  |  |  |  |  |

*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.**

o Comparación de rendimiento ARRAYLIST.

o Comparación de rendimiento LINKED\_LIST.

o Comparación de rendimiento para Insertion Sort.

O Comparación de rendimiento para Selection Sort.

o Comparación de rendimiento para Shell Sort.

o Comparación de rendimiento para MergeSort.

o Comparación de rendimiento para QuickSort.

# Maquina 2

## Resultados

ArrayList:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **T**amaño de la muestra (ARRAYLIST) | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 576.40 | 647.48 | 41.26 | 21.62 | 23.34 |
| 2000 | 2305.06 | 2587.87 | 77.07 | 52.64 | 50.90 |
| 4000 | 9515.19 | 10423.84 | 172.15 | 119.87 | 111.05 |
| 8000 | 39005.74 | 43526.40 | 414.71 | 281.61 | 242.50 |
| 16000 | 163392.00 | 180547.56 | 912.84 | 505.70 | 526.08 |
| 32000 | 653254.01 | 739160.51 | 2152.60 | 1192.05 | 1220.01 |

*Tabla 5. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.*

LinkedList:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **T**amaño de la muestra (LINKED\_LIST) | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 48643.41 | 42177.60 | 2276.68 | 1720.88 | 226.76 |
| 2000 | 376720.04 | 335093.46 | 10746.62 | 8427.45 | 892.70 |
| 4000 | 3297783.17 | 2905675.62 | 52287.14 | 38937.42 | 3575.72 |

*Tabla 6. 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 7. 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?

Con todos los algoritmos probados hay una relación entre lo teórico y lo que muestran las pruebas, pero hay una pequeña excepción, esto lo vimos con el MergeSort ya que en teoría siempre debería mostrar un comportamiento logarítmico, pero cuando fue probado en LinkedList mostró un comportamiento cuadrático.

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

Si hay diferencias entre los tiempos de ejecución de la máquina 1 y la máquina 2. La máquina 2 toma más tiempo en realizar las ejecuciones.

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

Seguramente estas diferencias se deben a los diferentes procesadores y a la diferencia de memoria RAM que tienen las máquinas.

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

El Arraylist ha mostrado ser el más rápido en cuanto a tiempo de ejecución de los 5 algoritmos con los que hemos realizado pruebas.

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.
2. Mergesort
3. Quicksort
4. ShellSort
5. InsertionSort
6. SelectionSort