OBSERVACIONES DE LA PRACTICA

Nicolás Maldonado Cod 201921739

David Rincón Cod 201921719

# Máquina 1 Máquina 2

|  |  |  |
| --- | --- | --- |
| **Procesadores** | Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz | Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz |
| **Memoria RAM (GB)** | 16GB | 8,00 GB |
| **Sistema Operativo** | Windows 10 64-bit | Windows 10 64-bit |

*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 0.0 640.62 0.0 31.25 15.62

2000 15.62 2640.62 62.50 62.50 31.25

4000 8734.37 10656.24 140.62 109.37 78.12

8000 36421.87 45906.24 109.37 218.75 171.87

16000 146765.62 198843.75 218.75 468.75 359.37

32000 +30min +30min 437,5 1015.62 843.75

64000 +30min +30min 4671.87 2218.87 1687.50

128000 +30min +30min 11203,12 4875.00 3640.62

256000 +30min +30min 29265.62 10312.50 7765.62

512000 SUPERO EL TAMAÑO DE LA LISTA

*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 32250.00 24359.37 281,25 1359.37 140.62

2000 264531.25 199343.75 1234.37 6156.25 531.31

4000 638436.76 543867.76 34406.25 24593.75 2109.37

8000 1487326.34 1362864.65 12109.37 108156.25 8453.12

16000 3867465.34 3437394.34 17453.12 471765.62 34531.25

32000 +30min +30min +30min +30min 48015.62

64000 +30min +30min +30min +30min 171531.25

128000 +30min +30min +30min +30min 749281.25

256000 +30min +30min +30min +30min +30min

512000 SUPERO EL TAMAÑO DE LA LISTA

*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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tamaño de la muestra Insertion (ARRAYLIST) Sort [ms]** | **Selection Sort [ms]** | **Shell**  **Sort**  **[ms]** | **Quick**  **Sort**  **[ms]** | **Merge**  **Sort**  **[ms]** |

1000 0.0 843.62 0.0 31.25 15.62

2000 19.74 4756.43 65.50 62.50 46.87

4000 10056.56 15656.24 133.63 125.00 78.12

8000 47843.28 68984.62 140.45 234.37 187.50

16000 194367.74 211856.74 274.74 500.00 406.25

32000 +30min +30min +30min 1062.50 843.75

64000 +30min +30min +30min 2343.75 1796.87

128000 +30min +30min +30min 5078.12 3843.75

256000 +30min +30min +30min 10781.25 8187.50

512000 SUPERÓ EL TAMAÑO DE LA LISTA

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

### Tamaño de la muestra Sort [ms]Insertion Sort [ms]Selection Shell[Sortms] Quick[Sortms] Merge[Sortms] (LINKED\_LIST)

1000 35240.00 34359.37 381,25 1531.25 156.25

2000 289521.25 329313.75 1454.43 7234.37 593.75

4000 632746.64 843957.75 38068.35 29171.87 2578.12

8000 1452846.26 2012486.37 29674.32 133234.37 9765.62

16000 3276395.45 4328495.13 27304.27 704921.87 6328.12

32000 +30min +30min +30min +30min 11556.62

64000 +30min +30min +30min +30min +30min

128000 +30min +30min +30min +30min +30min

256000 +30min +30min +30min +30min +30min

512000

*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. o

**Importante: Los valores en la graficas representan las operaciones que tardan menos de 10 minutos, aquellas con mas tiempo de procesamiento no se tienen en cuenta. (en otros casos, como el de comparación de rendimiento entre MergeSort y QuickSort, los datos del Array List estaban completos, pero se decidió no incluírlos para no arruinar la forma de la gráfica).**

* Comparación de rendimiento LINKED\_LIST.

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?

- Si, cada uno de los algoritmos estudiados generó resultados acordes a lo que se esperaba. De esta manera, Mergesort cumplió con las expectativas de ser el algoritmo más eficiente en cuanto al tiempo de ejecución debido a su complejidad la cual es n(log(n)). Por el otro lado, Quicksort, también demostró ser un algoritmo eficiente, sin embargo, es un poco más lento ya que puede, en el peor de los casos, tener una complejidad de O(n^2).

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

- Al tratarse de un programa el cual se basa en el ordenamiento de elementos en memoria, es muy importante tener en cuenta que cada máquina procesa los datos a una velocidad dependiente del procesador y aunque en el caso de las dos máquinas usadas en este caso la diferencia es mínima, los resultados al ejecutar pruebas de ordenamiento de datos en las maquinas siempre serán diferentes.

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

- La razón principal por la cual existen diferencias en los resultados se debe al rendimiento de procesamiento de cada máquina. De esta manera, una máquina con un mejor procesador mostrará mejores resultados en el tiempo de ejecución. Además, la manera en la que cada algoritmo funciona es diferente, por ejemplo, en Quicksort se requiere de una partición tras una distribución o “desorganización” de los datos y en el peor de los casos en el cual, el elemento seleccionado está al inicio o al final de la lista, el algoritmo es un poco más lento. Esto significa que la organización de los elementos en las listas afecta los resultados de tiempo de cada algoritmo.

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

-Mergesort es la estructura de datos mas eficiente al tener en cuenta solamente el tiempo.

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.

1. Mergesort

2. Shellsort

3. Quicksort

4. Insertionsort

5. Selectionsort