Departamento de Ingeniería de Sistemas y Computación

Estructuras de Datos y Algoritmos

ISIS 1225 – 202110

# LABORATORIO NO. 5: ORDENAMIENTOS RECURSIVOS

Julián Ricardo Villate Torres Cod 202020509

Sergio Pardo Gutiérrez Cod 202025720

## Objetivos

Comprender el funcionamiento y la implementación de los algoritmos de ordenamiento recursivos.

Al finalizar este laboratorio el estudiante estará en capacidad de:

* Comprender la complejidad (temporal y espacial) de los algoritmos de ordenamiento recursivos.  Desarrollar pruebas funcionales para validar los tiempos de ejecución de los ordenamientos recursivos.
* Comprender la diferencia entre ordenamientos iterativos y recursivos.
* Utilizar adecuadamente el ambiente de trabajo distribuido configurado.

Fecha Límite de Entrega

Máquina 1 Máquina 2

|  |  |  |
| --- | --- | --- |
| Procesadores | AMD A9-9425 RADEON R5 3.10 Ghz | Intel(R) Core(TM) i7-8565 CPU @1.80GHz 1.99GHz |
| Memoria RAM (GB) | 8 | 16 |
| Sistema Operativo | Windows 10 pro | Windows 10 Home 64 bits |

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

Estudiante 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort [ms] | Selection Sort [ms] | Shell Sort [ms] | Quick Sort [ms] | Merge Sort [ms] |
| 1000 | 1218 | 1203 | 46 | 15.625 | 62.5 |
| 2000 | 4021 | 4906 | 140 | 93.75 | 140.625 |
| 4000 | 25468 | 26906 | 468 | 218.75 | 251 |
| 8000 | 100687 | 99984 | 953 | 453.125 | 515.7 |
| 16000 |  |  |  | 921.875 | 1000 |
| 32000 |  |  |  | 2125.0 | 2390 |
| 64000 |  |  |  | 4843.75 | 4953.125 |
| 128000 |  |  |  | 10296.87 | 10437.5 |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

Tabla 2. Comparación del rendimiento para los ordenamientos iterativos y recursivos 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 | 102385 | 86125 | 4109 | 3843.75 | 609.375 |
| 2000 | 780796 | 674578 | 21796 | 15234.375 | 2031.25 |
| 4000 | 6342421 |  |  | 73078.125 | 8109.375 |
| 8000 |  |  |  | 310953.125 | 29687.5 |
| 16000 |  |  |  |  |  |
| 32000 |  |  |  |  |  |
| 64000 |  |  |  |  |  |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

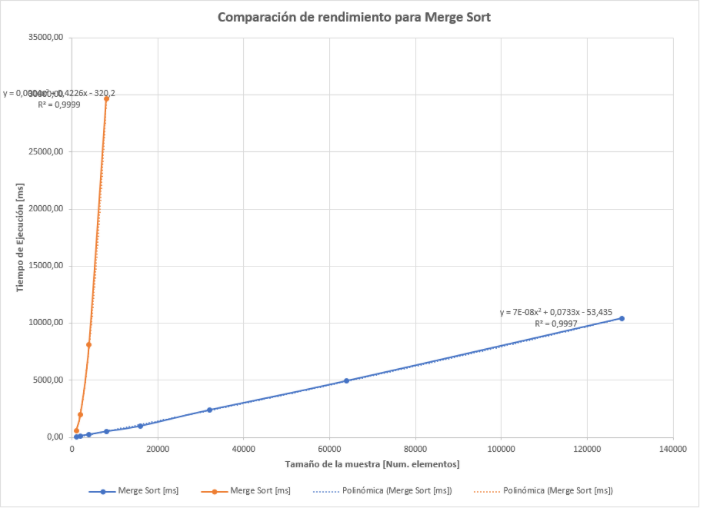
Tabla 3. Comparación del rendimiento para los ordenamientos iterativos y recursivos en la representación lista enlazada.

Algoritmo Arreglo (ARRAYLIST) Lista enlazada (LINKED\_LIST)

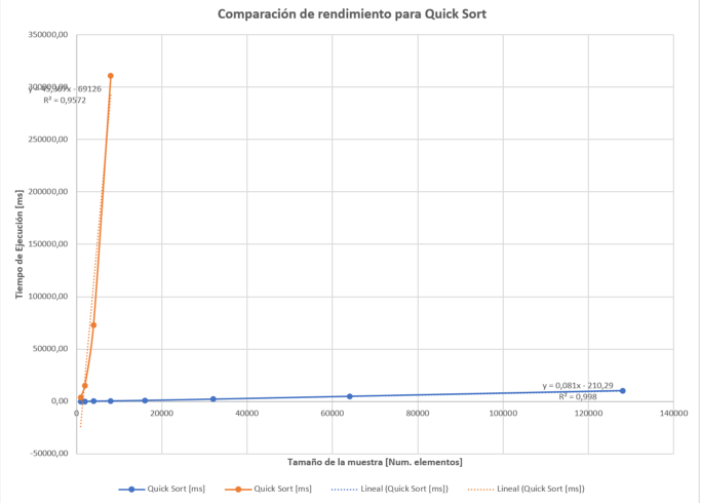
|  |  |  |
| --- | --- | --- |
| Mergesort |  | x |
| Quicksort | x |  |

Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

Grafica mergesort:



Grafica Quicksort:



Estudiante 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort [ms] | Selection Sort [ms] | Shell Sort [ms] | Quick Sort [ms] | Merge Sort [ms] |
| 1000 | 407.25 | 453.12 | 15.62 | 15.63 | 15.63 |
| 2000 | 1671.88 | 1937.5 | 46.88 | 46.88 | 46.88 |
| 4000 | 6921.88 | 7734.38 | 125 | 78.13 | 93.75 |
| 8000 | 29921.88 | 31875 | 281.25 | 156.25 | 239.58 |
| 16000 | 128156.25 | 135125 | 671.88 | 343.13 | 510.46 |
| 32000 | 516812.5 | 575296.88 | 1640.62 | 802.08 | 838.54 |
| 64000 | 2’173296.88 | 2’445812.5 | 4015.62 | 1625.4 | 1828.38 |
| 128000 |  |  | 10078.12 | 3390.63 | 3718.75 |
| 256000 |  |  | 23625 | 7625.32 | 7906.25 |
| 512000 |  |  | 37390.62 | 11953.13 | 12125.43 |

Tabla 2. Comparación del rendimiento para los ordenamientos iterativos y recursivos 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 | 32473.96 | 32578.13 | 1531.25 | 1296.88 | 156.25 |
| 2000 | 315328.13 | 282500 | 9000 | 5390.63 | 609.38 |
| 4000 | 2’740937.5 | 2’495437.5 | 44218.75 | 25421.88 | 2484.38 |
| 8000 |  |  | 200750 | 109656.25 | 10281.25 |
| 16000 |  |  | 996515.62 | 505890.63 | 42484.38 |
| 32000 |  |  |  | 2’183562.5 | 175421.88 |
| 64000 |  |  |  |  |  |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

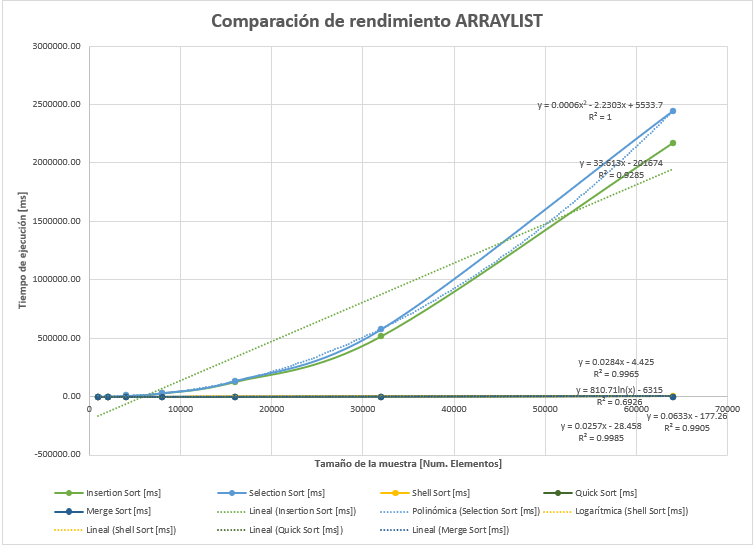
Tabla 3. Comparación del rendimiento para los ordenamientos iterativos y recursivos en la representación lista enlazada.

Algoritmo Arreglo (ARRAYLIST) Lista enlazada (LINKED\_LIST)

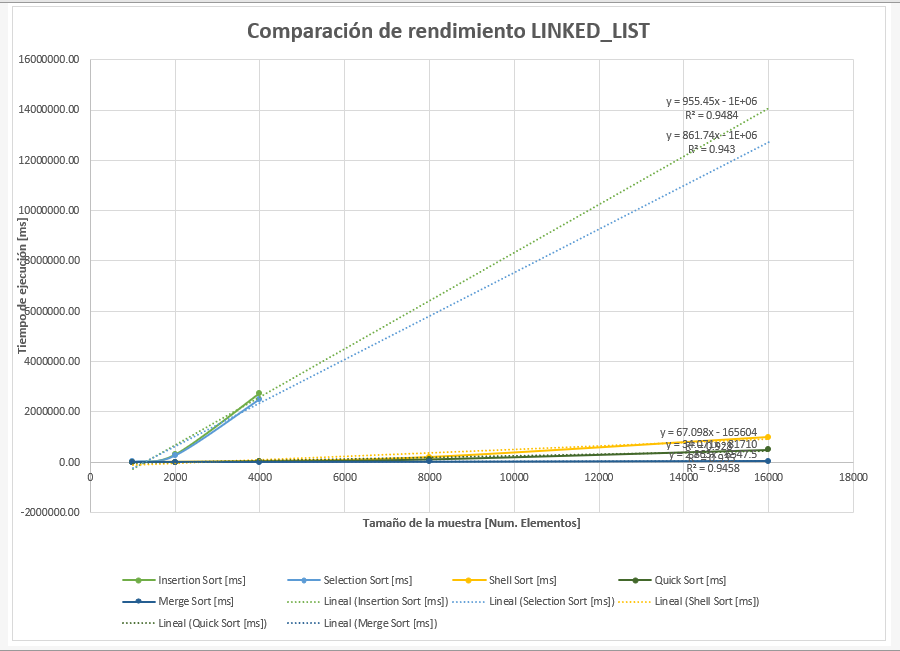
|  |  |  |
| --- | --- | --- |
| Mergesort |  | x |
| Quicksort | x |  |

Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

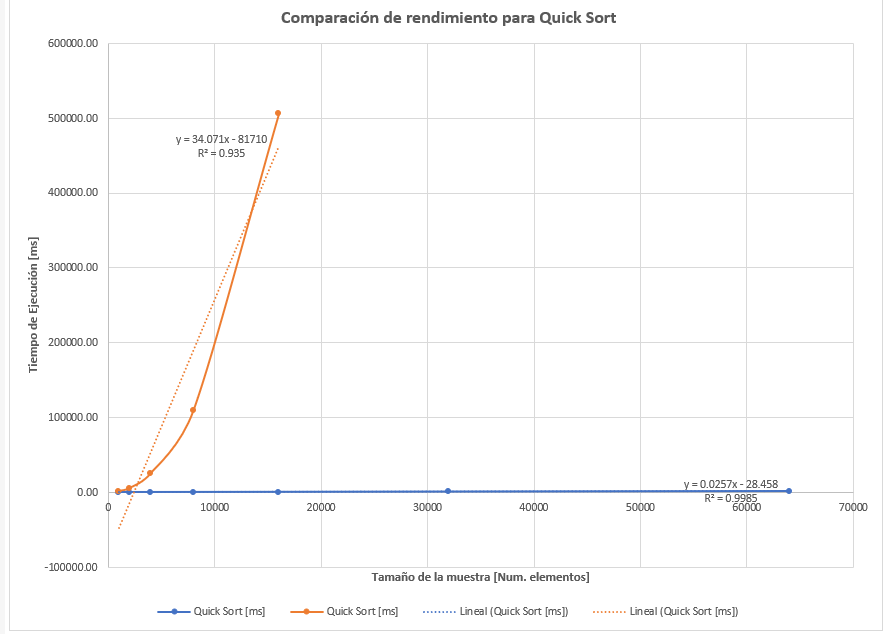
**Comparación de rendimiento ARRAYLIST.**



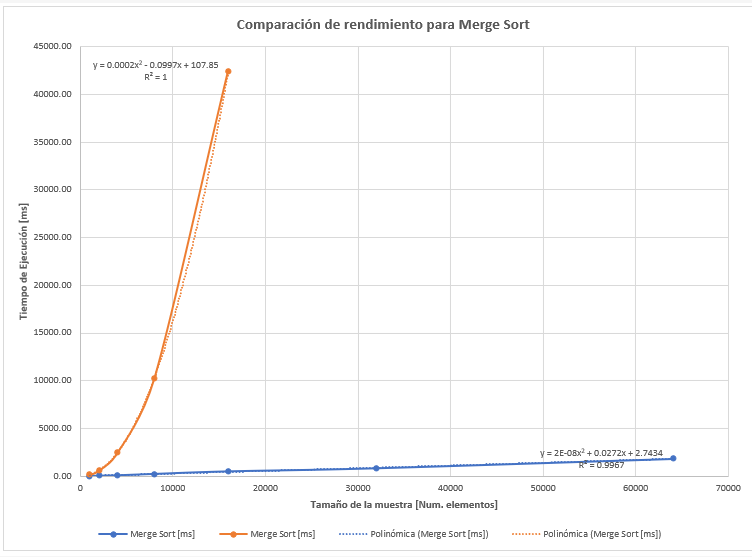
**Comparación de rendimiento LINKED\_LIST.**



**Comparación de rendimiento QuickSort.**



**Comparación de rendimiento MergeSort.**



Análisis

* + - ¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?

Si, como se esperaba se obtuvo un mayor rendimiento en el quicksort cuando se trata de un arreglo y en el mergesort cuando hablamos de una lista encadenada. Esto debido a la eficiencia del uso que requiere cada tipo de ordenamiento.

* + - ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?

Si, pero solo en los tiempos de ejecución. Los datos muestran en si las mismas tendencias.

* + - De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

Se deben principalmente a la diferencia de memoria RAM en cada equipo. Como se puede ver la maquina 2 tiene 16 GB de RAM mientras la maquina 1 solo tiene 8 GB

* + - ¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

Como en casos anteriores, pudimos notar un rendimiento general aplastantemente mayor en el arreglo.

* + - 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. Arraylist: Quicksort; Linked List: Mergesort.
2. Arraylist: Mergesort; Linked List: Quicksort.
3. Arraylist: Shellsort; Linked List: Shellsort.
4. Arraylist: Insertion sort; Linked List:Selection sort.
5. Arraylist: Selection sort; Linked List: Insertion sort.