OBSERVACIONES DE LA PRACTICA

Estudiante 1: Santiago Díaz Moreno Cod 201912247

Estudiante 2: Juana Mejia B Cod 202021512

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core(TM) i7-de 11va generation | 2,4 GHz Quad-Core Intel Core i5 |
| Memoria RAM (GB) | 4GB | 8 GB |
| Sistema Operativo | Windows 10 Pro 64-bits | MacOS Big Sur (Version 11.1) |

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 | 2078,125 | 2234,375 | 125,0 | 93.75 | 78.125 |
| 2000 | 8359,375 | 9437,5 | 281,25 | 281.25 | 156.25 |
| 4000 | 33843,75 | 37437,5 | 562,5 | 312.5 | 343.75 |
| 8000 | 135250,06 | 147640,625 | 1421,875 | 1156.25 | 796.875 |
| 16000 | Excede el tiempo de carga | Excede el tiempo de carga | 3171,875 | 1718.75 | 1578.125 |
| 32000 | Excede el tiempo de carga | Excede el tiempo de carga | 7562,5 | 3203.125 | 3281.25 |
| 64000 | Excede el tiempo de carga | Excede el tiempo de carga | 17750,0 | 7109.375 | 7156.25 |
| 128000 | Excede el tiempo de carga | Excede el tiempo de carga | 42984,375 | 14843.75 | 13843.75 |
| 256000 | Excede el tiempo de carga | Excede el tiempo de carga | 113578,125 | 34718.75 | 32390.62 |
| 512000 | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | Error | Error |

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 | 119765.625 | 106265.625 | 6093.75 | 3984.375 | 515.625 |
| 2000 | Excede el tiempo de carga | Excede el tiempo de carga | 27984.375 | 19921.875 | 2156.25 |
| 4000 | Excede el tiempo de carga | Excede el tiempo de carga | 128812.5 | 92750.0 | 8531.25 |
| 8000 | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | 425250.0 | 34750.0 |
| 16000 | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | 139093.75 |
| 32000 | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | 560171.875 |
| 64000 | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga |
| 128000 | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga |
| 256000 | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga |
| 512000 | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga | Excede el tiempo de carga |

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 | y = 9E-05x2 - 9,9536x + 89816 R² = 0,9865  O(n2) | y = 116159ln(x) - 902664 R² = 0,4469  O(log n) |
| Quick sort | y = 0,1333x - 521,74 R² = 0,994  O(n) | y = 0,0076x2 - 8,5881x + 5536,5 R² = 1  O(n2) |

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 | 635,68 | 713,65 | 43,84 | 31,42 | 32,45 |
| 2000 | 2704,81 | 2855,72 | 87,05 | 64,26 | 61,81 |
| 4000 | 10958,1 | 12822,03 | 189,49 | 134,78 | 128,86 |
| 8000 | 46997,26 | 55370,15 | 483,79 | 308,87 | 270,68 |
| 16000 | 228691,97 | 241021,5 | 1058,46 | 535,44 | 566,57 |
| 32000 | 856183,35 | 1039383,26 | 2554,27 | 1158,60 | 1201,29 |
| 64000 | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | 5759,54 | 2681,31 | 2561,03 |
| 128000 | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | 13617,37 | 5741,52 | 5518,88 |
| 256000 | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | 35799,87 | 13835,47 | 13110,88 |
| 512000 | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | 62202,01 | 31512,93 | 21906,89 |

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 | 46948,09 | 41990,67 | 2330,173 | 1652,54 | 244,06 |
| 2000 | 387986,04 | 387986,04 | 10965,92 | 9220,11 | 900,98 |
| 4000 | 3091533,48 | 2682711,97 | 249296,95 | 41611,57 | 4023,83 |
| 8000 | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | 182142,06 | 16233,47 |
| 16000 | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | 742944,06 | 63773,98 |
| 32000 | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | 276481,07 |
| 64000 | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor |
| 128000 | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor |
| 256000 | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor |
| 512000 | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor | El tiempo de carga exedio el maximo dicho por el profesor |

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 | y=0,0566x-482,39 R² = 0,9905 | y = 6E-08x2 + 0,0359x - 12,494 R² = 1 |
| Quick sort | y = 2E-07x2 + 0,0109x + 351,86 R² = 0,9947 | y = 0,003x2 - 0,9346x - 860,55 R² = 1 |

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.Chart, line chart

    Description automatically generated
  + Comparación de rendimiento LINKED\_LIST.Chart, line chart

    Description automatically generated
  + Comparación de rendimiento para Insertion Sort.Chart, line chart

    Description automatically generated
  + Comparación de rendimiento para Selection Sort.

Chart, line chart

Description automatically generated

* + Comparación de rendimiento para Shell Sort. Chart, line chart

    Description automatically generated
  + 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?

En algunos casos no se cumplio con el valor teorico, en los del estudiante 1, uno de los valores de merge sort dio como se esperaba pero el resto dieron una complejidad mayor. Los comportamientos de quicksort resutaron mucho más acordes a lo estupulado ya que se encuentran en el rango O(n^2) que fue lo que se enuncio teoricamente.

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

Nosotros creemos que las diferencias que existen se dan a cabo debido a diferencia de memoria RAM que tienen las diferentes maquinas, pues cuando existe menor memoria entonces se demora mayor tiempo en correr el programa de sorting.

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

Entre los resultados de las dos maquinas si existen diferencias en los tiempos que toma el computador para ordenar los datos, esto se debe a que las maquinas tienen memorias RAM diferentes, la que tiene menor memoria se demora más en correr el programa, pero el tiempo de las funcion.

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

Solo al tener el tiempo de ejcusion en cuenta se puede decir que Arraylist es más adecuada en cuanto a Quick y merge sort.

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.

El ranking es el siguiente:

1. Mergesort
2. Quicksort
3. Shellsort
4. Insertion
5. Selection

Nosotros hicimos este ranking en base a los resultados que obtubimos y al comparar los tiempos de prosesamiento de cada sort tanto en array\_list como single\_linked.