|  |  |  |
| --- | --- | --- |
|  | **Maquina 1** | **Maquina 2** |
| **Procesadores** | AMD Ryzen 5 3600 6-Core Processor, 3593 Mhz, 6 Core(s), 12 Logical Processor(s) | Intel i5-8300H (8) @ 4.000GHz |
| **Memoria RAM** | 16GB | 12GB |
| **Sistema operativo** | Microsoft Windows 10 pro 64-bits | Manjaro Linux 64-bit |

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

Maquina 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| **Small** | 769 | 442.71 | 36.46 | 36.46 | 36.46 |
| **10.00%** | 15009 | 163385.42 | 947.92 | 3703.13 | 614.58 |
| **20.00%** | 29490 | - | 1880.21 | 13031.25 | 1276.04 |
| **30.00%** | 43705 | - | 2916.67 | - | 2010.42 |
| **50.00%** | 71433 | - | 5317.71 | - | 3458.33 |
| **80.00%** | 111782 | - | 8359.38 | - | 5380.21 |
| **100.00%** | 138150 | - | 11125.00 | - | 6802.08 |

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

-Se demora mas en crear la muestra-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra**  **(LINKED\_LIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| **Small** | 769 | 12791.67 | 697.92 | 671.88 | 98.96 |
| **10.00%** | 15009 | \* | 557609.38 | 1030320.31 | 32718.75 |
| **20.00%** | 29490 | - | - | - | 126859.38 |
| **30.00%** | 43705 | - | - | - | 278447.92 |
| **50.00%** | 71433 | - | - | - | - |
| **80.00%** | 111782 | - | - | - | - |
| **100.00%** | 138150 | - | - | - | - |

*Tabla 3. Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada.*

*\*El Algoritmo se demoró más de dos horas y la prueba se detuvo*

*-Observación: El tiempo de creación de la muestra al usar LINKED\_LIST aumenta respecto a cuándo se usa ARRAYLIST.*

Maquina 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| **Small** | 769 | 403.53 | 31.83 | 67.9 | 26.34 |
| **10.00%** | 15009 | 158324.97 | 819.45 | 3622.56 | 551.82 |
| **20.00%** | - | - | - | - | - |
| **30.00%** | - | - | - | - | - |
| **50.00%** | - | - | - | - | - |
| **80.00%** | - | - | - | - | - |
| **100.00%** | - | - | - | - | - |

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

-Se demora mas en crear la muestra-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra**  **(LINKED\_LIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| **Small** | 769 | 14476.31 | 769.64 | 973.33 | 108.32 |
| **10.00%** | 15009 | \* | 516243.38 | \* | 35208.55 |
| **20.00%** | 29490 | - | - | - | - |
| **30.00%** | 43705 | - | - | - | - |
| **50.00%** | 71433 | - | - | - | - |
| **80.00%** | 111782 | - | - | - | - |
| **100.00%** | 138150 | - | - | - | - |

*Tabla 3. Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada.*

*-Observación: El tiempo de creación de la muestra al usar LINKED\_LIST aumenta respecto a cuándo se usa ARRAYLIST.*

* 1. Compare los resultados obtenidos con la complejidad teórica de cada algoritmo y con los resultados obtenidos de las pruebas en ambas maquinas respondiendo las siguientes preguntas:
  2. • ¿El comportamiento con relación al orden de crecimiento temporal de los algoritmos es acorde a lo enunciado teóricamente?

El comportamiento visto temporalmente es claramente relacionado con los datos experimentales obtenidos, exceptuando de QuickSort, pero se puede asumir que en este caso, QuickSort se acerco mas a su peor caso posible ( Cuadratico ) que a su caso promedio, por lo cual sigue siendo bastante pegado a los datos recibidos previamente

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

La differencia existia en terminos de numeros crudos, ya que las dos maquinas no van a ser exactamente iguales, sin embargo, el comportamiento general de el crecimiento temporal fue casi identico en general

* 1. • De existir diferencias, ¿a qué creen que se deben?

Como mencione previamente, las maquinas tienen differencias pequeñas, aun si tienen exactamente los mismos componentes, aparte de condiciones externas que generan differencias en los numeros totales.

|  |  |  |
| --- | --- | --- |
| **Algoritmo** | **Arreglo (ARRAYLIST)** | **Lista enlazada (LINKED\_LIST)** |
| **Insertion Sort** | O(n^2) | O(n^4) |
| **Shell Sort** | O(n^3/2) |  |
| **Merge Sort** | O(nlog(n)) | O(n^2) |
| **Quick Sort** | O(nlog(n)) | O(n^3) |

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

* ¿Cuál Estructura de Datos (ARRAY\_LIST o SINGLE\_LINKED) funciona generalmente mejor si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

En tiempo de ejecucion, ARRAY\_LIST rompe cualquier intento de LINKED\_LIST de intentar ser mas rapida, esto debido a su pesimo tiempo en lectura de elementos en la lista, que genera que mover elementos en general sea extremadamente lento.

* Teniendo en cuenta las pruebas de tiempo de ejecución reportadas por los algoritmos de ordenamiento probados (iterativos y recursivos), proponga un listado de estos ordenarlos de menor a mayor teniendo en cuenta el tiempo de ejecución que toma ordenar las obras de arte.

1. MergeSort

2. ShellSort

3. QuickSort

4. InsertionSort