OBSERVACIONES DE LA PRACTICA

Diego Alejandro Ramírez Garrido Cod 202013233

Sebastián Murcia Gómez Cod 202015229

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel inside core i3 | Intel(R) Core(TM) i7-8565U CPU @1.80GHz 1.99GHZ |
| Memoria RAM (GB) | 4 | 12 |
| Sistema Operativo | Windows 10 | Windows 10 pro x64 bits |

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 | 1500 | 1593.75 | 125 |  |  |
| 2000 | 6046.875 | 6390.625 | 218.75 |  |  |
| 4000 | 25078.125 | 27025 | 484.375 |  |  |
| 8000 | 104125 | 106156.25 | 1093.75 |  |  |
| 16000 | 535062.5 | 433703.125 | 2625 |  |  |
| 32000 | 1784671.875 | 1771875 | 6046 |  |  |
| 64000 |  | 7176093.5 | 13750 |  |  |
| 128000 |  |  | 33375 |  |  |
| 256000 |  |  | 86625 |  |  |
| 512000 |  |  | 136296.875 |  |  |

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 | 129046.875 | 114703 | 6390.25 |  |  |
| 2000 | 1057593.75 | 942046.875 | 30328.125 |  |  |
| 4000 |  |  | 156406.25 |  |  |
| 8000 |  |  |  |  |  |
| 16000 |  |  |  |  |  |
| 32000 |  |  |  |  |  |
| 64000 |  |  |  |  |  |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

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 | x |
| Quick sort |  |  |

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 | 2184,375 | 2078.125 | 125 | 62,5 | 78,125 |
| 2000 | 8656,25 | 80390.625 | 265 | 171,875 | 171,875 |
| 4000 | 36440,625 | 34578.125 | 562 | 406,25 | 390,625 |
| 8000 | 150734,375 | 148703.125 | 1437.5 | 921,875 | 828,125 |
| 16000 | 659968,75 |  | 3156.25 | 1671,875 | 1765,625 |
| 32000 | 2673343,75 |  | 7546.875 | 3593,75 | 3812,5 |
| 64000 |  |  | 18125 | 8125 | 8109,375 |
| 128000 |  |  | 20343.75 | 17375 | 17468,75 |
| 256000 |  |  | 128843.75 | 41890,625 | 39218,75 |
| 512000 |  |  |  | 86625 | 61312,5 |

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 | 109640.875 | 99703.75 | 5990.25 | 4390,625 | 687,5 |
| 2000 | 1037468.125 | 890640.875 | 27874.75 | 22187,5 | 2531,25 |
| 4000 |  |  |  | 106406,25 | 10218,75 |
| 8000 |  |  |  | 482359,375 | 40375 |
| 16000 |  |  |  |  | 163750 |
| 32000 |  |  |  |  | 660546,875 |
| 64000 |  |  |  |  |  |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

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 | x |
| Quick sort |  |  |

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.
  + 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?
   * + Efectivamente. En el caso de ArrayList, Quicksort y Mergesort se demoran mucho menos que el resto de algoritmos por ser de complejidad O(n log n), Shellsort tiene un comportamiento intermedio al ser O(n^3/2) e Insertion y Selection sort son los más lentos al tener un orden O(n^2). Igualmente, también es esperado que los ordenamientos en LinkedList se demoren más que en ArrayList gracias a que, para evaluar un valor dentro de la lista, tiene una complejidad O(n).
2. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?
   * + Diferencias menores, pero no significativas, especialmente en cuanto al comportamiento de las gráficas
3. De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?
   * + Diferencia en los procesadores de las máquinas, memoria RAM y procesos que pudieron ser efectivamente apagados en segundo plano. También está la diferencia del posible uso de la energía por parte de las diferentes máquinas.
4. ¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?
   * + Definitivamente Mergesort. Aunque Quicksort teóricamente puede ir más rápido que Mergesort, pero realmente depende de algo aleatorio, por lo cual puede ir más lento. Por otro lado, con el mismo número de datos, Mergesort siempre se demorará más o menos lo mismo.
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.
   * + Ranking (ascendente):
     + Insertion
     + Selection
     + Shell
     + Quick
     + Merge