OBSERVACIONES DE LA PRACTICA

Sergio Arango Arango Cod 201921814

Blanca Valencia Aguirre Cod 202013797

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel Core i7 | Intel Core I7 |
| Memoria RAM (GB) | 12 GB | 12 GB |
| Sistema Operativo | Windows 10 | Windows 10 |

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

# **Maquina 1**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) | Quick Sort (ms) | Merge Sort (ms) |
| 1000 | 557,63 | 666,32 | 41,91 | 35.65 | 42.97 |
| 2000 | 2364,67 | 2789,73 | 62,50 | 78.13 | 70.31 |
| 4000 | 8593,27 | 9981,83 | 153,38 | 128.91 | 117.19 |
| 8000 | 41789,71 | 40256,09 | 370,15 | 250.00 | 289.06 |
| 16000 | 161325,87 | 230776,98 | 874,83 | 535.16 | 503.91 |
| 32000 | 783197,76 | 1171897,28 | 1885,62 | 1031.25 | 1128.90 |
| 64000 | >30min | >30min | 4661,72 | 2480.47 | 2320.31 |
| 128000 | >30min | >30min | 11420,82 | 5027.34 | 5238.28 |
| 256000 | >30min | >30min | 31207,01 | 11246.09 | 10804.69 |
| 512000 | >30min | >30min | 49567,83 | 19097.66 | 17664.06 |

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 | 39343,125 | 38675,875 | 2171,663 | 2505.21 | 234.38 |
| 2000 | 375671,785 | 385421,875 | 10957,732 | 12796.87 | 828.13 |
| 4000 | 1786063,367 | >30min | 59000,654 | 32057.29 | 2989.13 |
| 8000 | >30min | >30min | 309251,542 | 173304.69 | 12958.33 |
| 16000 | >30min | >30min | 1354390,64 | 731250.56 | 52479.17 |
| 32000 | >30min | >30min | >30min | >30min | 254578.13 |
| 64000 | >30min | >30min | >30min | >30min | 1069078.13 |
| 128000 | >30min | >30min | >30min | >30min | >30min |
| 256000 | >30min | >30min | >30min | >30min | >30min |
| 512000 | >30min | >30min | >30min | >30min | >30min |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Insertion sort | X |  |
| Selection sort | X |  |
| Shell sort | X |  |
| 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.**
  + 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 Merge Sort.
  + Comparación de rendimiento para Quick Sort.

# **Maquina 2**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) | Quick Sort (ms) | Merge Sort (ms) |
| 1000 | 538,02 | 677,04 | 43,89 | 37,43 | 44,12 |
| 2000 | 2364,82 | 2790,12 | 61,45 | 76,34 | 69,8 |
| 4000 | 8592,98 | 9872,73 | 152,89 | 129 | 118,54 |
| 8000 | 41789,71 | 41460,43 | 370,15 | 251,2 | 270,3 |
| 16000 | 162325,31 | 231376,38 | 862,63 | 534,23 | 504,21 |
| 32000 | 782574,31 | 1174897,28 | 1874,21 | 1034,23 | 1112,8 |
| 64000 | >30min | >30min | 4731,73 | 2434,87 | 2345,40 |
| 128000 | >30min | >30min | 11541,91 | 5024,54 | 5212,90 |
| 256000 | >30min | >30min | 31434,42 | 11235,09 | 10814,23 |
| 512000 | >30min | >30min | 50127,97 | 19035,67 | 17625,68 |

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 | 39343,125 | 38675,875 | 2171,663 | 2510.23 | 212,38 |
| 2000 | 375671,785 | 385421,875 | 10957,732 | 12756,87 | 812,13 |
| 4000 | 1786063,367 | >30min | 59000,654 | 32347,29 | 2912,58 |
| 8000 | >30min | >30min | 309251,542 | 173404,69 | 52474,17 |
| 16000 | >30min | >30min | >30min | >30min | 254578,13 |
| 32000 | >30min | >30min | >30min | >30min | 1069458,13 |
| 64000 | >30min | >30min | >30min | >30min | >30min |
| 128000 | >30min | >30min | >30min | >30min | >30min |
| 256000 | >30min | >30min | >30min | >30min | >30min |
| 512000 | >30min | >30min | >30min |  |  |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Insertion sort | X |  |
| Selection sort | X |  |
| Shell sort | X |  |
| 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 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 Merge Sort.
  + Comparación de rendimiento para Quick Sort.

# **Preguntas de análisis**

1. ¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?

Sí, lo único sorprendente es que en las listas la diferencia entre selection e insertion fueron muy parecidas.

1. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?  
   En nuestro caso, teníamos los mismos computadores, por lo que la diferencia fue muy poca.
2. De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

Las pocas diferencias que hubieron, se deben a algunas aplicaciones abiertas en segundo plano que gastaban algo de RAM.

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

La estructura Arreglo es muchísimo mejor en términos de tiempo para realizar cualquier algoritmo de ordenamiento visto en la clase.

(Es posible que la Lista Encadenada sea mejor en términos de ocupación de memoria.)

1. Ordene los algoritmos por eficiencia de tiempo:
   * 1. Merge Sort
   * 2. Quick Sort
   * 3. Shell Sort
   * 4. Insertion Sort
   * 5. Selection Sort

Selection sort es claramente el peor y desde el laboratorio se sabe el órden por tiempo de los algoritmos iterativos. En estas pruebas los algoritmos recursivos (Merge y Quick) resultan ser mucho mejores que los algoritmos iterativos en términos de tiempo. Entre Merge y Quick son bastante parejos y no han una gran diferencia entre sustiempos pero Merge si parece ser un poco menos suscpetible a la alza de los tiempos cuando aumenta la cantidad de datos que debe manejar.