OBSERVACIONES DE LA PRACTICA

Estudiante 1:Juan Felipe García 202014961 jf.garciam1

Estudiante 2: Santiago Rodríguez 202020476 s.rodriguez64

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core (TM) i5-10300h CPU @2.50GHZ 2.50GHZ | Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50GHz |
| Memoria RAM (GB) | 8Gb | 12 GB |
| Sistema Operativo | Windows 10 64-bits | Windows 10 Home Single Language 64-Bits |

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) |
| 1000 | 1125,00 | 1343,75 | 62,50 |
| 2000 | 4671,88 | 5375,00 | 140,63 |
| 4000 | 18265,63 | 21531,25 | 312,50 |
| 8000 | 76390,63 | 87875,00 | 750,00 |
| 16000 | 292234,38 | 359734,38 | 1796,88 |
| 32000 | 1219656,25 | 1445171,88 | 4000,00 |
| 64000 | 4945706,10 | 5925204,70 | 10000,00 |
| 128000 | tiempo excesivo | tiempo excesivo | 24343,75 |
| 256000 | tiempo excesivo | tiempo excesivo | 59453,13 |
| 512000 | tiempo excesivo | tiempo excesivo |  |

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) |
| 1000 | 70109,38 | 62312,50 | 3734,38 |
| 2000 | 581312,50 | 510843,75 | 15890,63 |
| 4000 | 2324085,38 | 21097846,88 | 75359,38 |
| 8000 | 9528750,04 | 8312551,66 | 3763795,04 |
| 16000 | tiempo excesivo | tiempo excesivo | tiempo excesivo |
| 32000 | tiempo excesivo | tiempo excesivo | tiempo excesivo |
| 64000 | tiempo excesivo | tiempo excesivo | tiempo excesivo |
| 128000 | tiempo excesivo | tiempo excesivo | tiempo excesivo |
| 256000 | tiempo excesivo | tiempo excesivo | tiempo excesivo |
| 512000 | tiempo excesivo | tiempo excesivo | tiempo excesivo |

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 | Más eficiente | Menos eficiente |
| Selection sort | Más eficiente | Menos eficiente |
| Shell sort | Más eficiente | Menos eficiente |

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.
* Desde las primeras pruebas se nota que single linked toma mucho más tiempo que el array list. Debido a esto solo se pudieron tomar pocos datos, pues desde la revisión numero 4 ya tomaba demasiado tiempo y para terminarlo podía tomar días.
  + Comparación de rendimiento para Insertion Sort.
  + Comparación de rendimiento para Selection Sort.
  + Comparación de rendimiento para Shell Sort.

A partir de las anteriores graficas se nota que el metodo más efectivo es el que usa la lista encadenada y el metodo shell. Este tiene un tiempo mucho menor

# **Maquina 2**

## **Resultados**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) |
| 1000 | 625 | 859,38 | 46,88 |
| 2000 | 2588,54 | 3291,64 | 78,13 |
| 4000 | 9822,92 | 13031,25 | 187,5 |
| 8000 | 45604,17 | 57104,17 | 458,33 |
| 16000 | 188494,79 | 230677,08 | 1197,92 |
| 32000 | 812739,58 | 938901,04 | 2494,79 |
| 64000 | Exceso de tiempo | Exceso de tiempo | 6078,13 |
| 128000 | Exceso de tiempo | Exceso de tiempo | 14317,71 |
| 256000 | Exceso de tiempo | Exceso de tiempo | 35562,5 |
| 512000 | Exceso de tiempo | Exceso de tiempo |  |

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) |
| 1000 | 48828,13 | 43255,21 | 2609,38 |
| 2000 | 398729,17 | 371757,81 | 10687,5 |
| 4000 | 3223656.25 | 2846437,5 | 53473,96 |
| 8000 | Exceso de tiempo | Exceso de tiempo | 261890,63 |
| 16000 | Exceso de tiempo | Exceso de tiempo | 1241281.25 |
| 32000 | Exceso de tiempo | Exceso de tiempo | Exceso de tiempo |
| 64000 | Exceso de tiempo | Exceso de tiempo | Exceso de tiempo |
| 128000 | Exceso de tiempo | Exceso de tiempo | Exceso de tiempo |
| 256000 | Exceso de tiempo | Exceso de tiempo | Exceso de tiempo |
| 512000 | Exceso de tiempo | Exceso de tiempo | Exceso de tiempo |

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 | + mejor que linked | -peor que array |
| Selection sort | + mejor que linked | - peor que array |
| Shell sort | El mejor de todos | - peor que array peor que array |

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.

# **Preguntas de análisis**

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

Si, el orden de eficiencia en la práctica de peor a mejor es Selection, luego Insertion y finalmente Shell. En teoría Selection es O(n^2) siempre, Insertion varía entre O(n^2) y O(n) pero en promedio es O(n^2) y Shell tiene complejidad O(n^(3/2)) en promedio llegando a ser O(nlogn) en el mejor caso. Podemos notar como predicho que Insertion es un poco mejor que selection. También que los O pequeños de Shell es drásticamente mejor que los otros dos.

1. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas? Sí existe una clara diferencia. La maquina 2 lleva a cabo las operaciones en menor tiempo.
2. De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

Esta diferencia se debe a la diferencia entre las memorias RAM (8GB y 12GB) y al procesador. En ambos aspectos la maquina 2 supera a la 1. Sin embargo, en ambas situaciones las mediciones son consistentes entre si y con las predicciones.

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

ARRAYLIST es mejor que LINKED\_LIST en cada algoritmo, tanto en la teoría como en la práctica. Si se quisiera ser lo más eficiente se debe usar el algoritmo Shell.