OBSERVACIONES DE LA PRACTICA

Martín Rincón Cod 201914114

Mariana Ruiz Cod 202011140

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40 GHz | Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.19 GHz |
| Memoria RAM (GB) | 16GB | 8GB |
| Sistema Operativo | Windows 10 64-bits | Windows 10 Pro  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 | 718 | 1046.88 | 31.25 |
| 2000 | 2640.63 | 3500 | 78.125 |
| 4000 | 10750 | 13218 | 171.88 |
| 8000 | 45609.38 | 57796.88 | 500 |
| 16000 | 192265.63 | 224562.5 | 1515.625 |
| 32000 | 814968.75(>10min) | 919890.63 (>10min) | 2609.38 |
| 64000 |  |  | 6078.125 |
| 128000 |  |  | 15203.13 |
| 256000 |  |  | 32671.87 |
| 512000 |  |  |  |

*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 | 60531.25 | 59437.5 | 2515.63 |
| 2000 | 519828.13 | 405203.13 | 12390.63 |
| 4000 | (>10min) | (>10min) | 75015.24 |
| 8000 |  |  | 238156 |
| 16000 |  |  | 1317234.38 (>10min) |
| 32000 |  |  |  |
| 64000 |  |  |  |
| 128000 |  |  |  |
| 256000 |  |  |  |
| 512000 |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Insertion sort | X |  |
| Selection sort | X |  |
| Shell 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.

# **Maquina 2**

## **Resultados**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño de la muestra (ARRAYLIST) | Insertion Sort [ms] | Selection Sort [ms] | Shell Sort [ms] |
| 1000 | 709,38 | 869,79 | 46,88 |
| 2000 | 2809,38 | 3510,42 | 93,75 |
| 4000 | 11288,83 | 14057,29 | 203,13 |
| 8000 | 48583,33 | 57312,50 | 484,38 |
| 16000 | 189015,63 | 233781,25 | 1171,88 |
| 32000 | 773682,29 (>10 min) | 944520,83 (>10 min) | 2598,96 |
| 64000 |  |  | 6276,04 |
| 128000 |  |  | 15156,25 |
| 256000 |  |  | 35688,17 |
| 512000 |  |  |  |

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 | 46125,00 | 40619,79 | 2317,71 |
| 2000 | 367057,29 | 329187,50 | 9859,38 |
| 4000 | (>10 min) | (>10 min) | 47901,04 |
| 8000 |  |  | 220567,71 |
| 16000 |  |  | 1137750,00 (>10 min) |
| 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) |
| Insertion sort | X |  |
| Selection sort | X |  |
| Shell 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.

# **Preguntas de análisis**

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

Sí, el comportamiento es el esperado. Para justificar estos resultados es importante recordar que el algoritmo Shell tiene complejidad de O(n^3/2), mientras que Selection e Insertion tienen complejidad de O(n^2). Ambos algoritmos requieren de un uso intenso de la operación getElement(), la cual tiene complejidad de O(1) usando la estructura de datos Array(AL), mientras que en la Linked-list(LL), tiene complejidad O(n).

Por lo anterior, es de esperarse que implementando la estructura de datos Linked-List, cualquiera de los 3 algoritmos consuma más tiempo que si se usase la estructura de Array; lo cual se verifica en las graficas, los 3 algoritmos rinden mucho menos usando LL como estsructura de datos.

Como los datos usados estan desorganizados, la teoria nos dice que el tiempo de ejecucion de los 3 algoritmos debe crecer de manera polinomial respecto al tamaño de la lista a organizar. Al intentar con las diferentes regresiones, notamos que la que mejor se ajusta es la regresion polinomial; un polinomio de grado dos presenta un fit muy bueno en todos los casos, por lo que se valida la complejidad O(n^2) y O(n^3/2) de los algoritmos.

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

Sí, cada máquina ejecuta las pruebas en tiempos diferentes. Por ejemplo, en el caso de Insertion sort y de Selection sort, la máquina 2 ejecutaba las pruebas en menor tiempo. Ahora, en el caso de Shell sort, la máquina 1 era más rápida.

A pesar de que las 2 maquinas tenian cantidades de recursos muy diferentes, los tiempos que se tomaron los algoritmos no fueron siginificativamente distintos.

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

Las diferencias entra los tiempos de ambas maquinas no eran muy grandes; es posible que el OS de la maquina uno limitase los recursos destinados al proceso ya que, durante la ejecucion, la cpu no paso del 25% de uso, tampocó se usó más de la mitad de la RAM disponible.

Las ligeras varaiaciones entre los computadores, y entre los diferentes intentos en un mismo computador, se dan porque el sistema operativo tiene muchos subprocesos, los cuales cambian en el tiempo y afectan los tiempos de ejecucion de los algoritmos.

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

Array List