OBSERVACIONES DE LA PRACTICA

Andres Fernando Galvis Cod 201632930

Alejandro Palomino Cod 201913699

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD Ryzen 5 2600x | AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz |
| Memoria RAM (GB) | 16GB RAM | 8GB |
| Sistema Operativo | 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 | 817,708 | 796,875 | 812,5 |
| 2000 | 317,.688 | 3203,125 | 3203,125 |
| 4000 | 12625,00 | 12671,875 | 12796,875 |
| 8000 | 54192,70 | 54546,875 | 54359,375 |
| 16000 | 210687,500 | 210859,375 | 210671,875 |
| 32000 | se demoró mucho | se demoró mucho | se demoró mucho |
| 64000 | se demoró mucho | se demoró mucho | se demoró mucho |
| 128000 | se demoró mucho | se demoró mucho | se demoró mucho |
| 256000 | se demoró mucho | se demoró mucho | se demoró mucho |
| 512000 | se demoró mucho | se demoró mucho | se demoró mucho |

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 | 69765,625 | 71156,250 | 71484,375 |
| 2000 | 541515,625 | 542546,875 | 546171,875 |
| 4000 |  |  |  |
| 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) |
| Insertion sort |  |  |
| Selection sort |  |  |
| Shell 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.

# **Maquina 2**

## **Resultados**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) |
| 1000 | 901,04 | 906,24 | 906,25 |
| 2000 | 3661,453 | 3695,31 | 3598,96 |
| 4000 | 14749,99 | 14539,06 | 14354,17 |
| 8000 | 62947,91 | 62242,18 | 62979,16 |
| 16000 | 253197,91 | 253.07 | 245098,96 |
| 32000 | se demoró mucho | se demoró mucho | se demoró mucho |
| 64000 | se demoró mucho | se demoró mucho | se demoró mucho |
| 128000 | se demoró mucho | se demoró mucho | se demoró mucho |
| 256000 | se demoró mucho | se demoró mucho | se demoró mucho |
| 512000 | se demoró mucho | se demoró mucho | se demoró mucho |

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 | 75234.37 | 751250.00 | 74927.66 |
| 2000 |  |  |  |
| 4000 |  |  |  |
| 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) |
| Insertion sort |  |  |
| Selection sort |  |  |
| Shell 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.

# **Preguntas de análisis**

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

Lo que se puede observar más fácilmente son las 3 primeras graficas de las que se tienen los datos completos, ya que para linkedlist el tiempo de ejecución era muy largo y no se pudieron extraer mas datos y dado que con dos o un dato a pesar de que se haga regresión o regla de 3 no se ve realmente la tendencia nosotros optamos por no dejar los datos que obtuvimos del laboratorio. Ya que con dos datos la tendencia podría ser cualquiera se podría trazar una línea como una curva queda muy abstracto lo que se haga con los datos y lejos de la realidad.

De las 3 primeras graficas para lso dos computadores podemos decir que no se comportaron como teóricamente se deberían comportar, el selección por ejemplo que tiene un orden de O(n)

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

Existen diferencias en tiempo, al ejecutar las pruebas en las diferentes maquinas, la maquina 1 hace los procesos a una mayor velocidad que la maquina 2.

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

Las diferencias creemos nosotros que se pueden deber a las características de cada computador, ya que tienen diferentes especificaciones que hace que el proceso sea mas rápido y tome menos tiempo, como lo fue en el computador 1, que tiene mejores especificaciones que el 2.

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

Teniendo en cuenta únicamente los tiempos de ejecución de los algoritmos la mejor estructura de datos es el array, por que en general se nota en tan solo las dos primeras pruebas que la diferencia en tiempo es grande, mientras que array toma tiempos de 810-900 milisegundos en hacer una prueba con 1000 datos linked list se puede demorar 75000, lo cual es una diferencia muy grande que yo se lo atribuiría a el hecho de que tiene que crear las referencias de los datos.