OBSERVACIONES DE LA PRACTICA

Juan Sebastián Hoyos Cod 201822167

Lyda Acuña Caicedo Cod 201715885

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | 2.8 GHz Intel Core i7 cuatro nucleos | Intel® Celeron® N4000 CPU @ 1.10GHz 1.10GHz |
| Memoria RAM (GB) | 16.0 GB | 4.0 GB |
| Sistema Operativo | MacOs Catalina | Windows 10 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 | 824.747 | 1053.1300000000012 | 52,45399999999911 |
| 2000 | 2962.56799 | 4217,534000000001 | 131,55699999999928 |
| 4000 | 12127.606 | 17803,068999999996 | 249,83999999999895 |
| 8000 | 47966.304000000004 | 71454,451 | 604,93899999 |
| 16000 | 200248.45700000002 | 304556,3570000001 | 1309,8280000000013 |
| 32000 | 796716.0610000001 | + 900000 | 3085,4389999 |
| 64000 | + 900000 | + 900000 | 7321,009 |
| 128000 | + 900000 | + 900000 | 17537,403000000002 |
| 256000 | + 900000 | + 900000 | 45025,066 |
| 512000 | N/A | N/A | N/A |

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 | 50110,59899999998 | 44875,593 | 2518,2439999999815 |
| 2000 | 415620,202 | 366035,68200000003 | 11790,779999999926 |
| 4000 | + 900000 | + 900000 | 55753,153 |
| 8000 | + 900000 | + 900000 | 276104,6269999999 |
| 16000 | + 900000 | + 900000 | + 900000 |
| 32000 | + 900000 | + 900000 | + 900000 |
| 64000 | + 900000 | + 900000 | + 900000 |
| 128000 | + 900000 | + 900000 | + 900000 |
| 256000 | + 900000 | + 900000 | + 900000 |
| 512000 | + 900000 | + 900000 | + 900000 |

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 | 171807,625 | 232865,399 |
| Selection sort | 79816,900 | 205455,638 |
| Shell sort | 8368,6149 | 86541,700 |

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 | 939,2259 | 1172,32 | 54,326999999991 |
| 2000 | 3506,2100000009 | 4580,06666666666 | 149,97999999991 |
| 4000 | 15457,009999991 | 22275,305 | 255,07777777772 |
| 8000 | 50095,2800000001 | 75542,6000000009 | 624,043999999992 |
| 16000 | 230448,210009 | 335102,052 | 1532,48999999994 |
| 32000 | + 900000 | + 900000 | 3099,0069999992 |
| 64000 | + 900000 | + 900000 | 7483,006669 |
| 128000 | + 900000 | + 900000 | 17537,403 |
| 256000 | + 900000 | + 900000 | + 900000 |
| 512000 | N/A | N/A | N/A |

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 | 52401,0669999991 | 49168,3033333331 | 2253,259999991 |
| 2000 | 446120,0099999991 | 487254,634999999 | 10194,43999991 |
| 4000 | + 900000 | + 900000 | 53612,066999 |
| 8000 | + 900000 | + 900000 | 278209,283000009 |
| 16000 | + 900000 | + 900000 | + 900000 |
| 32000 | + 900000 | + 900000 | + 900000 |
| 64000 | + 900000 | + 900000 | + 900000 |
| 128000 | + 900000 | + 900000 | + 900000 |
| 256000 | + 900000 | + 900000 | + 900000 |
| 512000 | N/A | N/A | N/A |

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 | 59979,987 | 498521,076 |
| Selection sort | 87734,469 | 268211,468 |
| Shell sort | 3841,919 | 86067,263 |

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?**

Teoricamente un Arreglo es una matriz de un tamaño que varia dependiendo si se agregan o eliminan elementos, es ideal para obtener datos dentro de dicha matriz de manera rapida ya que cada dato tiene un unico espacio dentro de la matriz. Por otro lado, una Lista enlazada contiene diferentes elementos enlazados doblemente entre si, por lo cual a la hora de obtener un dato la Lista enlazada suele ser menos eficiente debido a que este tipo de estructura debe recorrer toda la lista para obtener dicho dato lo cual toma tiempo innecesario. Teniendo en cuenta lo anterior se puede evidenciar que se cumple la teoria debido a que la eficiencia de un Arreglo es mayor a la de una Lista enlazada para este caso en especifico en el que se requería obtener un dato.

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

Sí, existen diferencias de resultados entre las dos maquinas utilizadas.

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

Creemos que estas diferencias corresponden a factores como el sistema operativo, el espacio de memoria y el procesador de cada dispositivo, ya que obtuvimos resultados con mayor rapidez en la maquina “más rapida” y con caracteristicas superiores a la otra maquina en cuanto a los 3 puntos mencionados.

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

Si solo se tiene en cuanta los tiempos de ejecución la mejor estructura de datos es la de Arreglos (ArrayList) ya que como se muestra en la tabla 4 el tiempo es mucho menor que al utilizar una lista enlazada.