OBSERVACIONES DE LA PRACTICA

DIEGO ALEJANDRO GONZALEZ VARGAS Cod: 202110240

SEBASTIAN GUERRERO RIOS Cod: 202021249

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz | Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 3.40 GHz |
| Memoria RAM (GB) | 8,00 GB | 8,00 GB |
| Sistema Operativo | Windows 10 –x64 bits | Windows 10 |

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

# **Maquina 1**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| small | 768 | 390.625 | 31.25 | 31.25 | 31.25 |
| 10.00% | 15000 | 151328.125 | 765.5 | 2921.875 | 546.875 |

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (LINKED\_LIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| Small | 768 | 22744.79 | 1177.0 | 1026.04 | 140.625 |
| 3.33% | 5000 | NO ORDENO  (+30MIN) | 80411.46 | 107083.33 | 5979.33 |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* | 151328.125 | NO ORDENA(+30min) |
| *Shell Sort* | 765.5 | 80411.46 |
| *Merge Sort* | 546.875 | 5979.33 |
| *Quick Sort* | 2921.875 | 107083.33 |

Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

# **Maquina 2**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| Small | 768 | 406.25 | 31.25 | 31.25 | 31.25 |
| 10.00% | 15008 | 173062.50 | 796.88 | 2921.88 | 656.25 |

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (LINKED\_LIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| Small | 768 | 26067.71 | 1223.96 | 1479.17 | 154.92 |
| 10.00% | 15008 | N/A | 989125.00 | 1979015.63 | 60171.88 |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* | 173062.50 | N/A |
| *Shell Sort* | 796.88 | 989125.00 |
| *Merge Sort* | 656.25 | 60171.88 |
| *Quick Sort* | 2921.88 | 1979015.63 |

Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

# **Preguntas de análisis**

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

RTA/Para poder dar respuesta a esta pregunta se hace necesario el análisis diferenciado entre los resultados para las listas enlazas y para los arreglos:

En consecuencia, para las listas enlazadas se puede afirmar que sí hay una correspondencia teórica de los datos experimentales con los anexos teóricos de este laboratorio, pues como se puede observar, para el caso de ambas maquinas el algoritmo de Merge Sort fue con diferencia el más rápido en la tarea de ordenamiento. Así mismo, podemos observar que en concordancia con la gráfica 2 de la guía de laboratorio, los datos experimentales de ambas máquinas muestran al algoritmo Shell Sort como el segundo más eficaz, y a Quick Sort como el tercero. Adicionalmente, la ausencia de tiempos experimentales en ambas máquinas para el algoritmo Insertion apoya la gráfica de la guía de laboratorio mostrando la inviabilidad de su implementación.

Por otra parte, para el caso de Arraylist, los datos experimentales concuerdan con la teoría dada en los anexos de la guía de laboratorio. En este sentido, cabe resaltar como los tiempos algoritmos de Shell Sort, Merge Sort y Quick Sort son eminentemente más pequeños que los tiempos de Insertion Sort para ambas máquinas, lo que corresponde con la primera ilustración de la guía de laboratorio.

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

RTA/ Existe una pequeña diferencia en los tiempos de ejecución de la máquina 1 respecto de la máquina 2. A pesar de que para el caso de las listas enlazadas se manejaron lotes de datos diferentes, con la información de los arreglos se puede evidenciar una sutil diferencia en la que Máquina 1 es poco más rápida que la Máquina 2

1. De existir diferencias, ¿a qué creen que se deben?

RTA/ Tomando en consideración que todas las pruebas se realizaron con las máquinas ejecutando únicamente los ordenamientos, se considera que la diferencia no puede radicar en las tareas que ejecutaran cada uno de los equipos. Adicionalmente, las máquinas contaban con una Memoria RAM teóricamente igual (ya que se desconocen los datos de memoria utilizable), por lo que la diferencia no puede estar en esto. Finalmente, se considera que la pequeña diferencia de velocidades radica en la generación del procesador, ya que si bien ambos son Intel Core i7, para el caso de la máquina 1 es un procesador de séptima generación, mientras que para el caso de la máquina 2 el procesador es de 6 generación, lo que se considera tiene injerencia en la toma de tiempos de estas tablas.

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

RTA/ Como se puede evidenciar en las tablas de eficiencia (4) de los volúmenes más grandes evaluados en cada una de las máquinas, la estructura de datos más adecuada para alcanzar rapidez en el proceso de ordenamiento sería el arreglo ARRAY\_LIST, con una amplia diferencia sobre las listas enlazadas LINKED\_LIST

1. Teniendo en cuenta las pruebas de tiempo de ejecución por todos los algoritmos de ordenamiento estudiados (iterativos y recursivos), proponga un ranking de los mismo de mayor eficiencia a menor eficiencia en tiempo para ordenar la mayor cantidad de obras de arte.

|  |  |
| --- | --- |
| RANKING DE ALGORITMOS DE ORDENAMIENTO | |
| PUESTO | ALGORITMO |
| 1 | Merge Sort |
| 2 | Shell Sort |
| 3 | Quick Sort |
| 4 | Selection Sort |
| 5 | Insertion Sort |