OBSERVACIONES DE LA PRACTICA

# Samuel Josue Freire Tarazona, 202111460

# Jose David Martinez Oliveros, 202116677

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel Core i7 2630QM (2.0 Ghz) | AMD Ryzen 5 3450U with Radeon Vega Mobile Gfx. 2.10GHz |
| Memoria RAM (GB) | 16 GB | 12 GB |
| Sistema Operativo | **Windows 7 Professional (64 Bits)** | **Windows 10 (64 Bits)** |

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]** |
| 0.50% | 1000 | 77765.63 | 3656.25 | 2871.25 | 453.13 |
| 1.5% | 2000 | 489750.0 | 11046.88 | 12125.0 | 1296.88 |
|  | 4000 | - | 56421.88 | 56500.0 | 4890.63 |
|  | 8000 | - | 267000 | 342437.5 | 18062.5 |
|  | 16000 | - | 1216875 | - | 74875.0 |

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]** |
| 0.50% | 1000 | 56343.75 | 2453.13 | 2250.0 | 359.36 |
|  | 2000 | 475343.75 | 13046.88 | 11156.25 | 1546.88 |
|  | 4000 | - | 58437.5 | 58000 | 4703.13 |
|  | 8000 | - | 246281.25 | 373687.5 | 16484.38 |
|  | 16000 | - | 1204687.5 | - | - |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* | 283757.82 | 265843.75(mas efciente) |
| *Shell Sort* | 311000.00 | 304981.25(mas efciente) |
| *Merge Sort* | 103483.44(mas efciente) | 111273.44 |
| *Quick Sort* | 19915.63 | 5773.43(mas efciente) |

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]** |
| 0.50% | 1000 | 86828.13 | 4515.63 | 4671.88 | 546.88 |
| 100.00% | 2000 | 749125 | 16093.75 | 11296875 | 1375 |
|  | 4000 |  | 57421.88 | 59078.13 | 6875 |
|  | 8000 |  | 369203.13 | 529625 | 27000 |
|  | 16000 |  |  | 1452500 | 108828 |

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]** |
| 0.50% | 1000 | 52062.50 | 2453.13 | 2171.875 | 359.38 |
| 100.00% | 2000 | 424687.50 | 10625 | 11125 | 1484.38 |
|  | 4000 |  | 93781.25 | 91078.13 | 4328.13 |
|  | 8000 |  | 235243.75 | 367750 | 29265.63 |
|  | 16000 |  |  |  | 114703.13 |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* | 417976.56 | 238375(mas efciente) |
| *Shell Sort* | 111808.59 | 85525.78(mas efciente) |
| *Merge Sort* | 411434.38(mas efciente) | 118031.25 |
| *Quick Sort* | 215386.14 | 103872.40(mas efciente) |

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?

Como se puede ver en cada uno de los ordenamientos hay o grandes o mínimas diferencias, para este caso cada uno de los algoritmos cumple la orden de complejidad vista teóricamente. Por ejemplo, en la muestra de datos 2000 vemos que el resultado del orden Insertion O(N^2) se cumple ya que el valor del tiempo es uno muy superior a l de la muestra. Otro ejemplo lo podemos ver con el algoritmo Shell, la muestra de datos crece pero a un ritmo mucho más desacelerado, es decir se puede estar viendo afectado por el logaritmo de la función por lo que si cumple con su complejidad teórica que es e **O(N log 3 N).** Otro ejemplo lo vemos con el ordenamiento Quick, el cual, también crece a un ritmo mucho más desacelerado por el mismo caso anterior que se puede estar viendo afectado por el crecimiento logarítmico de los datos es decir, cumple con su complejidad teórica. Caso similar sucede con el algoritmo Merge, el cual al tener una complejidad parecida a las anteriores crece en un ritmo muy poco acelerado a comparación del Insertion.

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

Si en la maquina uno los resultado tiene a ser un poco más grandes, es decir, tienen acrecer más y tener un crecimiento más acelerado en términos del tiempo y de la cantidad de datos. Un ejemplo claro, es la columna de las dos tablas de Insertion, las cuales en la tabal 1 tiene un inicio más elevado y un crecimiento igualmente acelerado. Caso similar sucede en cada una de las otras columnas en donde vemos que en la maquina 2 tiene un desempeño, menos eficiente se podría decir que en la maquina 1 donde a pesar de no ser lo más rápido, es más eficiente.

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

Estas diferencias mencionadas anteriormente se puede genera por la RAM del computador. Ya que como vemos las diferencias entre las dos es muy notoria tenido la maquina 1 16 gigas de RAM y la maquina 2 12 de RAM. Otra casusa por la que se podría genera esto, es por el espacio, libre del computador, o por un factor de procesador porque como vemos el procesador de la máquina 1 es un poco más nuevo que el de la maquina 2.

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

En términos generales, y para las muestras de tamaño que hicimos. La mejor estructura e datos que se puede usar es el linked\_list ya que presenta números más pequeños en términos de las pruebas de sus algoritmos. A pesar de esto, la diferencia no es mucha, es decir, en algún punto la Array. Ya que esta puede llegar a marcar números más pequeños en los algoritmos de Merge que en el otro punto son muy malos.

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

El mejor: Merge Sort, es el que menos tiempo produce en la mayor cantidad de las pruebas

Luego: Quick

Luego: Shell

El peor: Insertion es el que mayor tiempo produce en la mayor cantidad de las pruebas