OBSERVACIONES DE LA PRACTICA

Estudiante 1 Sergio Rincón Cod 201914107

Estudiante 2 Luis Tejón Cod 202113150

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz | Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 2.61GHz |
| Memoria RAM (GB) | 16GB | 12GB |
| Sistema Operativo | Windows 10 home 64-bits | Windows 10 home 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]** |
| small | 294 | 593.75 | 109.375 | 125.0 | 93.75 |
| 10.00% | 13418 | 11140.625 | 12750.0 | 58718.7 | 6109.37 |

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 | 294 | 2218.75 | 296.87 | 343.75 | 125.0 |
| 10.00% | 1000 | 7406.25 | 4109.3 | 4468.75 | 718.75 |

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

|  |  |  |  |
| --- | --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |  |
| *Insertion Sort* | 593.75 | 2218.75 | Este fue el algoritmo más lento de todos |
| *Shell Sort* | 109.375 | 296.87 | Este algoritmo fue un poco mejor que Quick |
| *Merge Sort* | 93.75 | 125.0 | Merge fue el algoritmo más rápido de todos |
| *Quick Sort* | 125.0 | 343.75 | Quick fue un poco más lento que Shell, pero con tiempos similares |
|  | Se evidencia que los arreglos presentan tiempos de ejecución menores que los de las listas enlazadas | |  |

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 | 294 | 293.75 | 65.63 | 78.13 | 46.79 |
| 10.00% | 13418 | 5375.00 | 6239.59 | 29890.63 | 3083.34 |

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 | 294 | 1290.63 | 178.095 | 196.88 | 71.88 |
| 10.00% | 1000 | 5067.70 | 2724 | 2927.09 | 421.88 |

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

|  |  |  |  |
| --- | --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |  |
| *Insertion Sort* | 293.75 | 1290.63 | Este fue el algoritmo más lento en todos los casos |
| *Shell Sort* | 65.63 | 178.095 | Fue el segundo más rápido después de merge |
| *Merge Sort* | 46.79 | 71.88 | Fue el algoritmo más rápido en todos lo casos |
| *Quick Sort* | 78.13 | 196.88 | Fue un poco más lento que shell pero tiene tiempo muy parecido a este |
|  | En todos los casos los algoritmos fueron significativamente más rápido en los arreglos. | |  |

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?

Sí, cómo se mostró en clase los algoritmos de menor complejidad temporal mostraron una rapidez mayor en las dos muestras, y como se predecía la diferencias entre los algoritmos más rápidos con los más lento era mayor con muestras de mayor tamaño.

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

Sí. La diferencia es muy clara, sobretodo con respecto al procesador. En algunos casos los tiempos se duplican de una máquina a otra y ejecutando el mismo proceso.

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

Principalmente a los procesadores. La diferencia en cuanto a las operaciones que puede realizar un computador en un tiempo determinado varía, y por ende la rapidez con la que se ejecuta algoritmo varía también.

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

Los resultados dejan muy claro que para los algoritmos de ordenamiento que usamos son considerablemente mejores cuando se implementan a través de un TAD lista de tipo arreglo.

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.
2. Merge.
3. Shell. (aunque shell y quick estabán muy cerca para los datos que usamos fue mejor Shell, probablemente porque había varias fechas iguales.
4. Quick
5. Insertion.