OBSERVACIONES DE LA PRÁCTICA

Juan Camilo Prieto Avella, 201814815

Federico Melo Barrero, 202021525

# **Ambientes de pruebas**

|  | Máquina 1 | Máquina 2 |
| --- | --- | --- |
| Procesadores | 1,6 GHz Intel Core i5 de dos núcleos | Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz, 2401 Mhz, 2 procesadores principales, 4 procesadores lógicos. |
| Memoria RAM (GB) | 8.00 GB. | 4.00 GB. |
| Sistema Operativo | MacOS Big Sur 64-bits | Microsoft Windows 10 Home Single Language. |

*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 |  |  |  |  |  |
| 10.00% |  |  |  |  |  |

*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 |  |  |  |  |  |
| 10.00% |  |  |  |  |  |

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

| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| --- | --- | --- |
| *Insertion Sort* |  |  |
| *Shell Sort* |  |  |
| *Merge Sort* |  |  |
| *Quick Sort* |  |  |

*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 | 484.375 | 93.75 | 78.125 | 62.5 |
| 10.00% | 6656 | 220703.125 | 2562.5 | 7093.75 | 1625.0 |

*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]** |
| --- | --- | --- | --- | --- | --- |
| small | 294 | 1890.625 | 203.125 | 218.75 | 93.75 |
| 10.00% | 6656 | - | 4652.35 | 4323.22 | 3102.15 |

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

| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| --- | --- | --- |
| *Insertion Sort* | Más eficiente | Menos eficiente |
| *Shell Sort* | Más eficiente | Menos eficiente |
| *Merge Sort* | Más eficiente | Menos eficiente |
| *Quick Sort* | Más eficiente | Menos eficiente |

*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?

Los algoritmos se comportan como esperado en el sentido de que los que presentan mayor

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

Existe una diferencia notable entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas. La máquina 1 fue bastante más veloz que la máquina 2. Esto se puede deber a la diferencia en la memoria RAM de las máquinas.

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

Es probable que las diferencias existentes en los resultados obtenidos al ejecutar las pruebas en diferentes máquinas se deban a la memoria RAM de cada máquina. También pueden estar relacionadas con los procesos adicionales que en cada máquina estaban teniendo lugar mientras se hacían lasd pruebas. Se evidencia que una máquina con mayor memoria RAM responde de mejor manera a las pruebas.

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

Teniendo en cuenta únicamente los tiempos de ejecución de los algoritmos, funciona mejor el 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.

Se propone el siguiente ranking:

1. Merge Sort

3.

4.

5.