OBSERVACIONES DE LA PRACTICA

Camilo Esteban Mora Gómez - c.morag@uniandes.edu.co - 202112176

Karen Tatiana Vera Hernández - k.verah@uniandes.edu.co - 202113341

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core(TM) i3-10100 CPU @ 3.60GHz 3.60 GHz | 2,5 GHz Intel Core i5 de dos núcleos |
| Memoria RAM (GB) | 8GB | 10 GB |
| Sistema Operativo | Windows 10 64-bits | macOS catalina versión 10.15.7 |

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% | 768 | 265.63 | 52.08 | 15.62 | 15.63 |
| 10.00% | 15008 | 108890.62 | 27218.75 | 2078.12 | 171.87 |

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% | 768 | 15156.25 | 3093.75 | 671.87 | 109.37 |
| 10.00% | 15008 | Muy largo | Muy largo | Muy largo | 34531.25 |

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]** |
| 0.50% | 768 | 612.70 | 128.80 | 39.96 | 35.90 |
| 10.00% | 15008 | 240724.55 | 59786.13 | 4252.56 | 768.27 |

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% | 768 | 29030.87 | 5859.55 | 1313.27 | 198.96 |
| 10.00% | 15008 | Muy largo | Muy largo | Muy largo | 67630.90 |

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.

# **Preguntas de análisis**

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

-Si es acorde ya que su complejidad en insertsort es de O(n^2), shellsort es O(n^1.25), mergesort y quicksort en promedio son O(n(log(n)), auqnue quick sort en el peor caso puede ser O(n^2), estos promedios se ven reflejados con los resultados obtenidos en la tabla anterior.

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

-Si, existen diferencias.

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

-Principalmente se deben a que el procesador de la maquina 2 es de menos gigahercios, además el de la maquina 2 es de 2 núcleos mientas que la maquina 1 tiene 4 núcleos.

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

El mejor es Mergesort

A continuacion estan organizadas las estructuras desde el mas eficiente hasta el menos:

1.Mergesort

2.Quicksort

3.Shellsort

4.Insertionsort

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.

1.Mergesort

2.Quicksort

3.Shellsort

4.Insertionsort

5.Selectionsort

Selection es el menos eficiente ya que en todos los casos su complejidad es de o(n^2) a diferencia de insertion que en su mejor caso es o(n).