

# OBSERVACIONES DE LA PRACTICA

Jesed Alejandro Dominguez Piratova 202011992

Nicholas Barake Lacouture 202020664

## Ambientes de pruebas

| Máquina 1         |                                |  | Máquina 2                   |  |  |
|-------------------|--------------------------------|--|-----------------------------|--|--|
| Procesadores      | Intel i5 8th Gen (2.4ghz base) |  | AMD Ryzen 5 4500U (2.38GHz) |  |  |
| Memoria RAM (GB)  | 20GB                           |  | 8GB                         |  |  |
| Sistema Operativo | Windows 10 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] |
|--------------------------------|----------------------------------|---------------------|-----------------|-----------------|-----------------|
| 0.50%                          | 50                               | 15625,00            | 0,0             | 0,0             | 15625,00        |
| 100.00%                        | 10000                            | 432343,75           | 3125,0          | 13437,5         | 1906,25         |

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] |
|--------------------------------|------------------------------------|---------------------|-----------------|-----------------|-----------------|
| 0.50%                          | 50                                 | 31.25               | 31.25           | 15.625          | 0.0             |
| 100.00%                        | 10000                              | 133142187581234     | 295859.375      | 590281.25       | 410281.25       |

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

| Algoritmo      | Arreglo (ARRAYLIST) | Lista enlazada (LINKED_LIST) |
|----------------|---------------------|------------------------------|
| Insertion Sort | 432343,75           | 133142187581234              |
| Shell Sort     | 3125                | 295859                       |
| Merge Sort     | 13437,5             | 590281.25                    |
| Quick Sort     | 1906,25             | 410281.25                    |

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%                          | 50                               | 15.63               | 15.63           | 0.00            | 15.63           |
| 100.00%                        | 10000                            | 403484.38           | 5015.63         | 17046.88        | 3312.50         |

Tabla 5. 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%                          | 50                                 | 31.25               | 15.63           | 15.63           | 0.00            |
| 100.00%                        | 10000                              | 6754235.74          | 244532.23       | 67891.32        | 91430.00        |

Tabla 6. Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada.

| Algoritmo      | Arreglo (ARRAYLIST) | Lista enlazada (LINKED_LIST) |
|----------------|---------------------|------------------------------|
| Insertion Sort | 403484.38           | 6754235.74                   |
| Shell Sort     | 5015.63             | 244532.23                    |
| Merge Sort     | 17046.88            | 67891.32                     |
| Quick Sort     | 3312.50             | 91430.00                     |

Tabla 7. 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?  
Todos cumplen con lo enunciado teóricamente, a excepción del insertion sort. Nos parece algo supremamente peculiar, que a pesar de ser un algoritmo de ordenamiento estable sea el que más tarde en ordenar. En la toma de datos de 8000 llegó a demorar hasta más de 30 minutos en dar respuesta, en cambio los otros sorts no llegaron hasta ese punto. El insertion es el menos eficiente por lo que ordena los elementos uno por uno.
- 2) ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?  
Sí, existen diferencias al tomar distintos datos con la misma estructura y el mismo tamaño de muestra. Como se puede evidenciar en las tablas, los tiempos varían drásticamente.
- 3) De existir diferencias, ¿a qué creen que se deben?  
Se cree que estas diferencias se deben a la diferencia de procesador y de memoria RAM. Entre mejores sean estas características, la prueba se realizará de manera más optima. Pero también pueden entrar en juego cuanto poder de procesamiento de la máquina se esté utilizando en el algoritmo, si se tiene otro programa abierto, demora más.
- 4) ¿Cuál Estructura de Datos funciona mejor si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?  
Si únicamente vemos los tiempos, el mejor algoritmo es el merge sort porque utiliza el concepto de divide y conquista. Sin importar lo grande que sea la lista, este algoritmo lo parte de mitad en mitad hasta que no se pueda más, y luego coge esas sublistas y las une de manera ordenada.
- 5) 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.  
De mejor a peor: Merge sort, Shell sort, Quick sort, Insertion Sort.