OBSERVACIONES DE LA PRÁCTICA

Simon Peña Alarcon Cod 202512907

Manuel Santiago Figueroa Cod 202511697

Juan Sebastian Chacón Cod 202513196

# **Ambientes de pruebas**

|  |  |
| --- | --- |
|  | Computador 1 |
| Procesador | Intel core I5 |
| Memoria RAM (GB) | 16 GB |
| Sistema Operativo | Windows 11 |

Para realizar las siguientes pruebas, es importante que las ejecuten utilizando el archivo de datos **large**, ya que es con este conjunto donde realmente se puede observar el comportamiento y la eficiencia de los algoritmos en contextos más exigentes.

# **Ordenamientos Iterativos**

## **Resultados para ordenamientos iterativos con Array List**

|  |  |  |  |
| --- | --- | --- | --- |
| Porcentaje de la muestra | Insertion Sort  (Array List) | Selection Sort (Array List) | Shell Sort (Array List) |
| 0.50% |  |  |  |
| 5.00% |  |  |  |
| 10.00% |  |  |  |
| 20.00% |  |  |  |
| 30.00% |  |  |  |
| 50.00% |  |  |  |
| 80.00% |  |  |  |
| 100.00% |  |  |  |

Tabla 2. **Resultados en computador 1 para ordenamientos iterativos con Array List.**

## **Resultados para ordenamientos iterativos con Linked List**

|  |  |  |  |
| --- | --- | --- | --- |
| Porcentaje de la muestra | Insertion Sort  (Linked List) | Selection Sort  (Linked List) | Shell Sort (Linked List) |
| 0.50% |  |  |  |
| 5.00% |  |  |  |
| 10.00% |  |  |  |
| 20.00% |  |  |  |
| 30.00% |  |  |  |
| 50.00% |  |  |  |
| 80.00% |  |  |  |
| 100.00% |  |  |  |

Tabla 3. **Resultados en computador 1 para ordenamientos iterativos con Single Linked List.**

# **Ordenamientos Recursivos**

## **Resultados para ordenamientos recursivos con Array List**

|  |  |  |
| --- | --- | --- |
| Porcentaje de la muestra | Merge Sort (Array List) | Quick Sort (Array List) |
| 0.50% |  |  |
| 5.00% |  |  |
| 10.00% |  |  |
| 20.00% |  |  |
| 30.00% |  |  |
| 50.00% |  |  |
| 80.00% |  |  |
| 100.00% |  |  |

Tabla 4. **Resultados en computador 1 para ordenamientos recursivos con Array List.**

## **Resultados para ordenamientos recursivos con Linked List**

|  |  |  |
| --- | --- | --- |
| Porcentaje de la muestra | Merge Sort (Linked List) | Quick Sort (Linked List) |
| 0.50% |  |  |
| 5.00% |  |  |
| 10.00% |  |  |
| 20.00% |  |  |
| 30.00% |  |  |
| 50.00% |  |  |
| 80.00% |  |  |
| 100.00% |  |  |

Tabla 5. **Resultados en computador 1 para ordenamientos recursivos con Single Linked List.**

Una vez alla llenado l

# **Comparación de tiempo mejores algoritmos de ordenamiento**

**Complete esta sección únicamente cuando se le indique en las instrucciones. En este punto, deberá comparar el tiempo de ejecución entre el mejor algoritmo de ordenamiento recursivo y el mejor algoritmo de ordenamiento iterativo, los cuales debió haber identificado en el punto anterior.**

**Por favor, registre las mediciones en el espacio correspondiente de la tabla, especificando claramente:**

* **El nombre de cada algoritmo utilizado**
* **El tipo de estructura de datos empleada (ArrayList o SingleLinkedList)**

**Asegúrese de completar todos los campos solicitados en la tabla e incluir los nombres según corresponda, no es necesario que vuelva a correr las pruebas esta información la puede sacar de las tablas anteriores.**

## **Computador 1**

|  |  |  |
| --- | --- | --- |
| Porcentaje de la muestra | Algoritmo recursivo | Algoritmo iterativo |
| 0.50% |  |  |
| 5.00% |  |  |
| 10.00% |  |  |
| 20.00% |  |  |
| 30.00% |  |  |
| 50.00% |  |  |
| 80.00% |  |  |
| 100.00% |  |  |

Tabla 6. **Resultados en computador 1 mejores algoritmos**

# **Preguntas de análisis parte 1**

# **¿Cómo varía el comportamiento de cada algoritmo con respecto al tamaño de los datos? RTA: El tiempo de ejecución aumenta a medida que crece el tamaño de la muestra. En Insertion y Selection Sort el crecimiento es cuadrático (O(n²)), mientras que en Shell Sort se observa una curva más suave, cercana a O(n log n). Para Merge y Quick Sort la tendencia es casi lineal–logarítmica, lo que corresponde a su complejidad teórica O(n log n).**

# **¿Qué diferencias observas en el rendimiento de un mismo algoritmo al utilizar ArrayList frente a SingleLinkedList? RTA: En general, los algoritmos son más rápidos con ArrayList porque el acceso a elementos es directo (O(1)). En cambio, en LinkedList las operaciones de acceso requieren recorrer nodos (O(n)), lo que penaliza el tiempo en muestras grandes.**

# **¿Qué algoritmo iterativo mostró el mejor comportamiento general? RTA: Shell Sort fue el más eficiente en ambas estructuras, con tiempos mucho menores a Insertion y Selection. Mientras Insertion y Selection se degradan rápidamente, Shell mantiene un crecimiento más lento y estable en todas las muestras.**

# **¿Qué algoritmo recursivo presentó el mejor desempeño general? RTA: Quick Sort se comportó mejor que Merge Sort en la mayoría de tamaños, mostrando menor tiempo de ejecución y una línea de tendencia más baja. Merge fue estable, pero Quick Sort fue claramente más rápido.**

# **Preguntas de análisis parte 2**

**¿Cuál de los dos algoritmos seleccionados como “mejores” es más eficiente al crecer la muestra?  
RTA: Quick Sort. Su crecimiento se ajusta mejor a O(n log n), mientras que Shell, aunque eficiente, depende del intervalo de reducción y puede acercarse a cuadrático en casos desfavorables. Quick Sort es más escalable.**

**¿Qué estructura de datos fue más favorable?  
RTA: En ambos casos, ArrayList fue superior a LinkedList. La ventaja se mantiene sobre todo en muestras grandes, donde el acceso secuencial de LinkedList se vuelve un cuello de botella.**

**¿Qué ventajas prácticas tendría el algoritmo ganador?  
RTA: Quick Sort en ArrayList sería ideal para sistemas que manejan grandes volúmenes de datos (bases de datos, análisis en memoria, motores de búsqueda) porque garantiza rapidez y escalabilidad.**

**Si la lista estuviera parcialmente ordenada, ¿cambiaría tu elección?  
RTA: Sí. En ese escenario, Insertion Sort mejora notablemente su rendimiento (cercano a O(n)), por lo que podría ser preferible en listas casi ordenadas. Sin embargo, para datos generales no estructurados, Quick Sort sigue siendo la mejor opción**