



Implementación y Análisis de Algoritmos de Búsqueda y Ordenamiento en Java

Objetivos:

- Implementar diferentes algoritmos de búsqueda y ordenamiento en Java.
- Analizar la complejidad temporal de los algoritmos utilizando la notación Big O.
- Comparar el rendimiento de los algoritmos mediante pruebas y benchmarks.
- Documentar hallazgos y conclusiones en un informe estructurado.

Pasos a Seguir:

1. Implementación de Algoritmos:

- **Búsqueda:**
 - Implementa el algoritmo de búsqueda secuencial.
 - Implementa el algoritmo de búsqueda binaria (requiere que los datos estén ordenados).
- **Ordenamiento:**
 - Implementa el algoritmo de ordenamiento burbuja.
 - Implementa el algoritmo de ordenamiento por inserción.
 - Implementa el algoritmo de ordenamiento por selección.

2. Análisis de Complejidad:

- Para cada algoritmo implementado:
 - Describe el pseudocódigo o la lógica de funcionamiento.
 - Utiliza la notación Big O para analizar la complejidad temporal en el mejor caso, peor caso y caso promedio.
 - Justifica tus análisis explicando cómo llegaste a cada conclusión de complejidad.

3. Comparación de Rendimiento:

- Diseña conjuntos de datos de prueba (pequeños, medianos y grandes).
- Realiza pruebas cronometradas para medir el tiempo de ejecución de cada algoritmo con estos conjuntos de datos.
- Registra los tiempos de ejecución y analiza cómo varían con diferentes tamaños de datos.



- Utiliza gráficos o tablas para visualizar los resultados si es posible.

4. Documentación y Reporte:

- Documenta tu código con comentarios claros que expliquen cada paso y cada método utilizado.
- Escribe un informe que incluya:
 - Introducción al problema y objetivos del proyecto.
 - Explicación detallada de cada algoritmo implementado con ejemplos de código.
 - Análisis de la complejidad Big O para cada algoritmo.
 - Resultados de las pruebas de rendimiento, incluyendo gráficos o tablas si los hubiera.
 - Conclusiones sobre el rendimiento relativo de los algoritmos y sus aplicaciones prácticas.
 - Recomendaciones para la selección de algoritmos en diferentes escenarios basadas en tus hallazgos.

5. Entrega a través de Moodle:

- **Subida del Código:**
 - Crea un archivo comprimido (.zip) con todo el código fuente y los archivos necesarios.
 - Asegúrate de incluir un archivo README con instrucciones para ejecutar los programas.
 - Subir el archivo comprimido a la sección de tareas del Moodle designada para esta actividad.
- **Subida del Informe:**
 - Prepara un documento en formato PDF con el informe completo.
 - Asegúrate de que el informe esté bien estructurado y cubra todos los puntos mencionados anteriormente.
 - Subir el archivo PDF a la misma sección de tareas de Moodle.
- **Entrega de la Presentación:**
 - Prepara una presentación en PowerPoint o PDF con los puntos clave de tu informe y los resultados más importantes.
 - Subir la presentación a la sección de tareas de Moodle antes de la fecha de entrega.

Recursos Sugeridos:

- Entorno de Desarrollo Integrado (IDE) para Java (como IntelliJ IDEA, Eclipse).
- Bibliotecas de Java para mediciones de tiempo de ejecución.



- Plataforma de colaboración para documentación y presentación de informes (Google Docs, Markdown).

Notas Importantes:

- Asegúrate de entender cada algoritmo antes de implementarlo. Investiga y practica con ejemplos simples antes de abordar la implementación completa.
- Mantén una comunicación constante con tu equipo para discutir avances, desafíos y resultados obtenidos.
- Revisa y valida tu trabajo antes de la entrega para asegurar que todos los aspectos estén cubiertos y documentados correctamente.