



Tecnológico de Monterrey

Campus Guadalajara

Nombre del trabajo:

E2. Actividad Integradora 2

Estudiantes:

Yair Salvador Beltrán Ríos

Daniel Alfredo Barreras Meraz

Matrícula:

A01254673

A01256805

Análisis y diseño de algoritmos avanzados

Fecha de entrega:

02 de diciembre

La implementación del sistema de análisis de transmisiones y detección de códigos maliciosos ha sido un ejercicio valioso que demuestra la importancia de seleccionar los algoritmos apropiados para problemas específicos de búsqueda y análisis de patrones. Las decisiones tomadas en la implementación reflejan un balance entre eficiencia computacional y claridad del código.

El uso del algoritmo de Rabin-Karp para la búsqueda de subcadenas fue una decisión particularmente acertada. Como se explica en los comentarios del código, se eligió este algoritmo por varias razones fundamentales:

1. Su complejidad promedio de $O(n + m)$ lo hace eficiente para buscar múltiples patrones en textos largos, lo cual es crucial cuando se analizan múltiples códigos maliciosos en distintas transmisiones.
2. La adaptabilidad del algoritmo permitió modificarlo para trabajar específicamente con caracteres alfanuméricos (A-Z, 0-9), optimizándolo para el caso de uso específico.
3. Su capacidad para manejar colisiones de hash de manera efectiva mediante verificaciones adicionales garantiza la precisión en la detección.

Para el problema de encontrar palíndromos, la implementación del algoritmo de expansión alrededor del centro con complejidad $O(n^2)$ demuestra un compromiso interesante entre simplicidad y eficiencia. Aunque existen algoritmos más sofisticados como Manacher's con complejidad $O(n)$, la claridad y mantenibilidad del código fueron factores importantes en esta decisión.

La búsqueda de la subcadena común más larga mediante programación dinámica con complejidad $O(m*n)$ representa un enfoque robusto que garantiza encontrar la solución óptima. La estructura de la matriz dp facilita no solo encontrar la longitud de la subcadena común más larga sino también sus posiciones exactas en los archivos.

Una de las lecciones más valiosas de esta implementación fue la importancia de la documentación clara y el manejo eficiente de archivos. El uso de funciones modulares como `readFile()` con complejidad $O(n)$ simplifica la lectura de datos y hace el código más mantenible.

Las áreas de oportunidad para futuras mejoras incluyen:

1. Implementación de procesamiento paralelo para analizar múltiples transmisiones simultáneamente.
2. Optimización de memoria en la búsqueda de subcadenas comunes para archivos muy grandes.
3. Incorporación de técnicas de detección de patrones más avanzadas para identificar variantes de códigos maliciosos.

En conclusión, este proyecto ha sido un excelente ejercicio en el diseño de algoritmos y estructuras de datos, demostrando cómo diferentes técnicas algorítmicas pueden combinarse para resolver un problema complejo de análisis de datos. La experiencia ha reforzado la importancia de considerar no solo la eficiencia teórica sino también la practicidad y mantenibilidad del código en aplicaciones del mundo real.