



Instituto Tecnológico y de Estudios Superiores de Monterrey.

Campus Monterrey

Escuela de Ingeniería y Ciencias

Análisis y diseño de algoritmos avanzados (Gpo 570). TC2038.570

E1. Actividad Integradora 1

Reflexión

Dirigido a:

Felipe Castillo Rendón

Por:

Oscar Arreola Alvarado | A01178076

**6 de Octubre del 2024
Monterrey, Nuevo León.**

Reflexión

En este proyecto tuve la oportunidad de aprender acerca de una variedad de algoritmos y de cómo podrían ser utilizados para resolver problemas de la vida real y de alto impacto. Considero que una de las partes más importantes, a la hora de aprender algoritmos, es la parte de buscar áreas de aplicación ya que esto muestra un propósito claro detrás de todo el esfuerzo que implica la implementación y búsqueda de nuevos algoritmos eficientes.

Para poder solucionar la evidencia, fue necesario llevar a cabo la implementación de 3 diferentes algoritmos, cada uno resolviendo una de las problemáticas especificadas por la evidencia.

En primer lugar, fue necesario implementar KMP para encontrar, de manera eficiente, si alguno de los códigos maliciosos se encontraba en las transmisiones. Para este problema, la complejidad computacional final fue de $O(n + m)$, una complejidad mucho mejor a la que se podría obtener usando el algoritmo naive de $O(n * m)$. La complejidad de KMP es lineal ya que hace uso de un preprocesamiento que permite determinar, de manera eficiente, la posición actual del patrón a buscar, sin necesidad de regresar al inicio en caso de que el siguiente carácter del patrón no sea igual al siguiente carácter de donde se está realizando la búsqueda.

Por otro lado, para resolver el segundo problema, el de encontrar el palíndromo más grande, se usó el algoritmo de Manacher, el cual permite obtener la solución en un tiempo de $O(n)$. De nuevo, esta complejidad es mucho mejor a la obtenida por el algoritmo naive $O(n^2)$ y demuestra la gran diferencia que existe al usar algoritmos eficientes. El algoritmo funciona al guardar la longitud de los palíndromos y reutilizar esta información para calcular la longitud de los siguientes palíndromos.

En tercer lugar, se utilizó el algoritmo LCS para encontrar el substring común más grande entre los archivos de transmisión. La implementación utilizada tiene una complejidad de $O(n * m)$ y hace uso de dp para calcular los resultados de manera eficiente. Para cada posición, considera que el substring más grande tiene de tamaño 1 + el tamaño del substring mayor que puede ser formado sin considerar ese carácter. Algo que cabe destacar es que la solución solo ocupa $O(m)$ de espacio ya que se hizo una optimización donde solo se guarda la última fila del dp, ahorrando así memoria considerable.

Finalmente, puedo decir que aprendí acerca de bastantes algoritmos nuevos que sirven para resolver problemas de manera eficiente. Las ideas detrás de los algoritmos resultan bastante interesantes y definitivamente inspiran a buscar algoritmos eficientes para otros problemas.