ITESM CAMPUS GUADALAJARA
Aldo Alejandro Degollado Padilla A01638391
Reflexión individual, Actividad Integradora 1
TC2038.1
17/09/2021

En la primera actividad integradora del curso, se nos introdujo a la situación problema uno: transmisiones de datos comprometidas. En dicho problema se nos plantea un escenario en el que debemos de encontrar un patrón de código malicioso que está tomando cierto control de algunos dispositivos. Se nos dice que nosotros ya tenemos identificado el patrón y se hace la pregunta de si sería posible encontrarlo en diferentes segmentos de código. Y es aquí, donde, dejando de lado la parte narrativa del problema, llegamos al tema de la actividad: comparación de strings.

A lo largo de esta actividad, se nos pidió implementar tres distintos tipos de algoritmos que nos ayudaran a:

1. Identificar un patrón en una cadena de caracteres, para evitar un patrón que ponga en riesgo la transmisión.
2. Identificar el palíndromo más largo de una cadena de caracteres, para evitar el código espejado.
3. Encontrar la misma y más larga subcadena de caracteres entre dos cadenas de caracteres.

Para resolver estos 3 retos, se nos pidió que nuestro programa recibiera 5 archivos; 3 con un patrón particular (mcode) y 2 con texto general (transmission).

El primer ejercicio decidimos resolverlo utilizando el algoritmo de KMP, el cual nos permitía encontrar coincidencias de un patrón dentro de una cadena de caracteres, sin necesariamente tener que pasar por todos los elementos. Este algoritmo tiene una complejidad de O(m) y fue por esto mismo que decantarnos por él.

El segundo problema lo resolvimos utilizando el algoritmo de Manacher. En este caso nos decantamos por este algoritmo debido a que ya lo conocíamos y sabíamos lo eficiente que llega a ser al momento de encontrar palíndromos. Este

algoritmo tiene una complejidad media de O(n), aunque en el peor de los escenarios, esta puede llegar a transformarse en O(n^2).

Por último, pero no menos importante, se nos pidió que encontráramos el patrón más largo que pudiéramos encontrarnos al comparar dos cadenas de caracteres distintas. En este ejercicio fue en el que más batallamos debido a que no teníamos conocimiento previo de un algoritmo que pudiera ayudarnos a resolver este problema, por lo que decidimos comenzar a experimentar. Este proceso de codificación derivó en un algoritmo de complejidad O(m*n), con el cual decidimos mantenernos. Este algoritmo guarda los subcadenas de caracteres más largas y al final compara entre ellos cuál es el mayor.

Esta actividad nos permitió aplicar los conocimientos adquiridos en las primeras 5 semanas de este curso, y a la vez nos ayudó a expandir nuestro conocimiento a distintos tipos de algoritmos.