Imagen que contiene firmar

Descripción generada automáticamente

Análisis y Diseño de Algoritmos Avanzados

12 de septiembre de 2021

Luis Armando Salazar Lopez

ITESM

A01114901

Actividad Integradora 1: Reflexión

**Reflexión**

Me parece muy interesante y curioso la manera en la que se puede llegar a detectar viruses y amenazas desde el momento de revisar los interiores de un archivo para evitar problemas causados por código mal intencionado.

Para el primer parte del problema, necesitábamos un algoritmo capaz de encontrar cadenas de caracteres dentro de archivos, esta función sirve cuando se sabe de un código mal intencionado que ya es conocido y puede encontrarse de una manera más fácil para evitar los problemas que pueda causar, en este caso para reducir la complejidad del algoritmo, optamos por usar Knuth-Morris-Pratt.

En lo personal me enfoqué en la segunda parte de la situación problema, la cual consistía en encontrar el palíndromo más grande dentro de una cadena de caracteres, para reducir grandemente la complejidad utilicé el algoritmo de Manacher el cual encuentra el palíndromo más grande de manera lineal.

Para el tercer ejercicio utilizamos programación dinámica para encontrar la subcadena más larga que comparten dos cadenas de caracteres.

Lo que me pareció más interesante de esta actividad fue el intentar encontrar un error que ocurría aleatoriamente, para encontrar el error tuve que repasar detalladamente el código para así poder encontrar el por qué a veces funcionaba bien el código y otras veces fallaba. Al final termine encontrando que era debido a un vector que se le indicaba un parámetro incorrecto, por ejemplo vector[-2], esto y la combinación de un min() causó que a veces fallara el código debido a que vector[-2] regresa un valor inesperado que a veces era positivo o negativo de gran valor. Al encontrar el error bastó con utilizar un “if statement” para que no volviera a ocurrir.

En general la solución de esta situación problema fue bastante interesante y desafiante, al tener que aprender más sobre distintas formas de resolver las partes de esta actividad al igual que utilizamos aprendizajes anteriores y los que hemos obtenido en lo que llevamos de la materia.

Complejidades:

Lectura de archivo de texto: Complejidad: O(n) donde n es el número de caracteres del archivo.

Algoritmo 1: Complejidad: O(n + m) donde n y m son los caracteres de ambas cadenas que se recibieron de parámetro.

Algoritmo 2: complejidad O(n), n es el tamaño del string.

Algoritmo 3: Complejidad: O(m\*n) donde m y n son las longitudes de ambos archivos de transmisión.