

E1. Actividad Integradora 1

El algoritmo principal que implementamos es KMP para buscar los patrones entre los diferentes archivos de texto, este algoritmo de búsqueda compara carácter por carácter de izquierda a derecha, pero cada vez que no coincide, utiliza una tabla preprocesada para omitir la comparación de caracteres durante la coincidencia. Decidimos usar KMP ya que su complejidad en el peor de los casos es de $O(n)$, lo que representa una gran ventaja de tiempo al ser más rápido en comparación con otros algoritmos de búsqueda.

Para la tercera parte de encontrar el substring más largo, usamos programación dinámica y es que de esta manera podemos obtener el substring común más largo con una complejidad de tiempo de $O(m*n)$ ya que optimiza el espacio al tener una variable para almacenar el punto final de la subcadena común más larga en la cadena X y otra variable para almacenar la longitud de la subcadena común más larga.

Estos algoritmos nos permitieron dar solución a la situación problema, demostrando que se puede encontrar código malicioso en los dos archivos de transmisión al encontrar un patrón de substrings que representan que los dos archivos han sido intervenidos.

Referencias

GeeksforGeeks. (2022, 15 julio). *Print the longest common substring*. Recuperado 1 de octubre de 2022, de <https://www.geeksforgeeks.org/print-longest-common-substring/>

Mathur, T. (2021, 15 noviembre). *KMP Algorithm | Knuth Morris Pratt Algorithm*. Scaler Topics. Recuperado 1 de octubre de 2022, de <https://www.scaler.com/topics/data-structures/kmp-algorithm/>

GeeksforGeeks. (2022b, septiembre 30). *KMP Algorithm for Pattern Searching*. Recuperado 1 de octubre de 2022, de <https://www.geeksforgeeks.org/kmp-algorithm-for-pattern-searching/>