Tecnológico de Monterrey

E1. Actividad Integradora 1

Aaron Hernandez Jimenez	A01642529
Jorge Antonio Arizpe Cantú	A01637441

07 de Septiembre del 2024

Análisis y diseño de algoritmos avanzados (Gpo 602)

Tecnológico de Monterrey

Tecnológico de Monterrey

Documento de Reflexión

1. KMP (Knuth-Morris-Pratt)

- **Descripción**: El algoritmo KMP busca un patrón dentro de una cadena utilizando una tabla de prefijos que evita la comparación redundante de caracteres. La idea clave es aprovechar el prefijo más largo que coincide con el sufijo de la parte ya comparada del patrón.
- **Aplicación en el código**: En este caso, el KMP se utiliza para buscar patrones dentro de las transmisiones (`pattern_search()`) y devolver las posiciones de coincidencia.
- **Complejidad computacional**: El KMP tiene una complejidad de O(n + m), donde **n** es la longitud de la cadena y **m** es la longitud del patrón. Esto lo hace más eficiente que una búsqueda tradicional que sería O(n*m).

2. Manacher's Algorithm

- **Descripción**: Este algoritmo se utiliza para encontrar el palíndromo más largo en una cadena en tiempo lineal. Convierte la cadena original en una forma donde cada carácter está separado por un símbolo especial para simplificar el manejo de palíndromos de longitud par e impar.
- **Aplicación en el código**: Manacher se utiliza en la función encontrar_palindromo_mas_largo() para localizar el palíndromo más largo dentro de las transmisiones.
- **Complejidad computacional**: El algoritmo de Manacher tiene una complejidad de O(n), donde **n** es la longitud de la cadena. Esta es una mejora significativa en comparación con un enfoque de fuerza bruta que tendría una complejidad de $O(n^2)$.

3. Longest Common Subsequence (LCS)

- **Descripción**: El algoritmo LCS encuentra la subsecuencia más larga que dos cadenas tienen en común. Se utiliza programación dinámica para construir una tabla que almacena las longitudes de subsecuencias comunes.

- **Aplicación en el código**: LCS se implementa en la función find_longest_common_sequence() para encontrar la subsecuencia más larga entre dos transmisiones.
- **Complejidad computacional**: El algoritmo LCS tiene una complejidad de O(m*n), donde **m** y **n** son las longitudes de las dos cadenas comparadas. Aunque este es un algoritmo cuadrático, sigue siendo una opción aceptable cuando se trata de cadenas relativamente cortas.