Tecnológico de Monterrey

E1. Actividad Integradora 1

TC2038.602: Análisis y diseño de algoritmos avanzados (Gpo 602)

29.10.2023

—

Equipo 1

David Medina Domínguez, A01783155
Arantza Guadalupe Parra Martínez, a01782023
Luis Carlos Rico Almada, A01252831

Reflexión y documentación: Investigación sobre diferentes **algoritmos**.

Para esta entrega hemos decidido usar los siguientes algoritmos, que se explicaran a continuación junto con sus especificaciones, porque decidimos utilizarlos, su complejidad computacional, ventajas, desventajas y como planeamos implementarlos.

Búsqueda de subcadenas: Knuth-Morris-Pratt (**KMP**)

Busca una subcadena dentro de un texto utilizando una tabla que indica cuántos caracteres podemos saltar al buscar una coincidencia, y nos evita comprobaciones innecesarias.

Complejidad: O(n)

Ventajas: No retrocede en el texto y procesa la subcadena una vez.

Desventajas: La preparación inicial puede ser más lenta en comparación con otros algoritmos.

Búsqueda de palíndromos: **Manacher**

Puede utilizar la información de palíndromos previamente detectados para evitar comprobaciones redundantes.

Complejidad: O(n)

Ventajas: Rápido y encuentra todos los palíndromos en un texto.

Desventajas: Complejo de implementar.

Subcadena más larga común: **Suffix Array y LCP**

Esta estructura de datos es una lista ordenada de todos los sufijos de una cadena.

Complejidad:
Construccion: O(n)
Busquedas: O(log n)

Ventajas: Menor consumo de memoria en comparación con árboles de sufijos. Permite operaciones eficientes de búsqueda y comparación de subcadenas.

Desventajas: Limitaciones para trabajar con textos grandes y complejidad de almacenamiento por tamaño de arreglos

Reflexión Final

La solución de esta evidencia podría ser realizada utilizando distintos algoritmos vistos en clase. Por lo que, la correcta selección, bajo nuestro criterio, de los algoritmos a utilizar fue el primer paso para su solución.
Para la identificación de texto espejado, el algoritmo de Manacher era una sencilla, dado que es una herramienta específicamente diseñada para encontrar palíndromos en cadenas. Por lo que, fue una decisión directa.

Por otro lado,para identificar códigos maliciosos dentro de una transmisión había más opciones. Aunque la función Z podría haber sido una elección viable, decidimos usar KMP debido a nuestra familiaridad con el algoritmo y nuestra confianza en adaptarlo a las particularidades del problema.

La decisión sobre cómo solucionar la búsqueda del string más largo fue más compleja. Ya que, al comenzar a definir cuál iba a ser el algoritmo seleccionado, notamos que ninguno de los algoritmos por sí solos, lograban realizar una búsqueda completa dentro de los dos archivos para encontrar el string más largo. El limitante recaía en que las búsquedas de “patrones” sólo se realizaban al inicio del string, por lo que, si el patrón se encontraba en otra posición o en otro “formato”, no eran encontrados por el algoritmo.

Para esto, se encontraron 2 alternativas: la primera consistía en utilizar un arreglo de sufijos junto con la matriz de prefijos común más larga (LCP array). LCP es una estructura de datos que almacena los prefijos comunes más largos entre 2 sufijos consecutivos del arreglo de sufijos.
La otra alternativa encontrada fue utilizar KMP dentro del arreglo de sufijos. Donde, cada uno de los sufijos, de Transmisión y transmisión 2 eran utilizados como “patrón” y “texto” respectivamente.
Al final, se decidió utilizar la primera opción pues consideramos que era la solución más eficiente. Esto, porque a través de investigación descubrimos que fue creado para realizar búsquedas de substrings.

Por último, la realización de esta evidencia nos enseñó la importancia de conocer una variedad de algoritmos y técnicas. Nos ayudó a darnos que más allá de conocer los algoritmos, se debe de saber cuándo y cómo aplicarlos o combinarlos de manera efectiva. La correcta selección de algoritmos se magnifica cuando se trata de lidiar con información verdadera. Donde, un error no solo lleva a resultados incorrectos, sino que compromete la seguridad y confidencialidad de los datos manejados. Nos gustaría enfatizar que las soluciones propuestas pueden tener vulnerabilidades o "espacios" para errores. Por ello, este código puede ser optimizado para solucionar vulnerabilidades actuales sobre todo en manejo de datos de grandes cantidades y para mejorar la eficacia del código.