Reflexión E1. Actividad Integradora 1.

Vicente Jesús Ramos Chávez A01750996

El código presentado aborda tres problemas clave relacionados con el análisis y procesamiento de texto en archivos: la detección de coincidencias de patrones, la búsqueda del palíndromo más largo y la identificación de la subcadena común más larga entre dos transmisiones. Cada uno de estos problemas se resuelve utilizando algoritmos fundamentados en técnicas eficientes de programación, mostrando una comprensión sólida de las estructuras y métodos necesarios para abordar este tipo de desafíos.

En la primera parte, el código utiliza una función basada en el método find para buscar patrones específicos (contenidos en los archivos de "mcode") dentro de las transmisiones. Este enfoque permite determinar si un patrón existe en una transmisión y, de ser así, devuelve su posición. Aunque esta técnica es directa y sencilla, su complejidad computacional, O(n·m), puede volverse costosa en textos largos o con patrones grandes. Este es un punto de optimización posible, donde algoritmos más avanzados como Knuth-Morris-Pratt (KMP) o Boyer-Moore podrían reducir significativamente el tiempo de ejecución.

La segunda parte del código se centra en encontrar el palíndromo más largo dentro de las transmisiones mediante programación dinámica. Este enfoque utiliza una matriz bidimensional para rastrear las subcadenas que cumplen la condición de ser palíndromos, asegurando que se evalúen todas las combinaciones posibles de inicio y fin en el texto. Aunque el algoritmo tiene una complejidad de O(n^2), lo que es aceptable para textos moderadamente largos, en casos más extensos podría beneficiarse de técnicas optimizadas como el método de "expandir desde el centro", que simplifica el análisis reduciendo tanto el espacio como el tiempo de procesamiento.

Por último, en la tercera parte, el algoritmo para encontrar la subcadena común más larga entre dos transmisiones utiliza nuevamente programación dinámica. Este enfoque rellena una tabla que rastrea el progreso de coincidencias entre los dos textos, logrando identificar el segmento compartido más extenso. La complejidad de O(n·m) es adecuada para entradas de tamaño medio, aunque en escenarios donde el tiempo sea crítico, estructuras como árboles de sufijos podrían ofrecer una solución más rápida.

En general, el código refleja un dominio de los principios básicos de algoritmos aplicados a problemas de texto. Cada sección está diseñada para resolver un problema concreto con claridad y eficiencia dentro de sus límites, aunque hay oportunidades claras de optimización mediante técnicas avanzadas. Este ejercicio no solo evidencia el valor de los algoritmos clásicos en programación, sino también la importancia de evaluar sus limitaciones y explorar mejoras, especialmente cuando se enfrentan datos más grandes o contextos en tiempo real.