

Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática

CI-0117 Programación Paralela y Concurrente

Grupo 003

Prof: Francisco Arroyo

Idea Proyecto #4

Estudiantes:

Agustín Soto Chaves C4K199

Andrés Salas C37104.

Noviembre 2025

Descripción de la idea

El algoritmo propone explorar la relación entre dos cadenas construyendo subsecuencias de forma incremental a partir de cualquier posición inicial, analizando solo la información que aparece hacia adelante en cada una, la idea central consiste en que al ubicarse en dos posiciones i y j de las cadenas originales, se observan los caracteres que aparecen en los substrings que comienzan desde esas posiciones. Cuando una letra aparece en ambos substrings, se abre una ruta de exploración donde el algoritmo avanza en ambas cadenas hasta la primera aparición de esa letra y la agrega a una subsecuencia en construcción; luego continúa el proceso desde ese nuevo punto, repitiendo la comparación hacia adelante para agregar más coincidencias. Este algoritmo se repite para todos los puntos de inicio, de modo que el algoritmo analiza rutas alternativas cuando existen múltiples coincidencias potenciales en un mismo paso.

Funcionamiento

El procedimiento inicia eligiendo un punto de partida en ambas cadenas, desde esos índices el algoritmo identifica qué letras aparecen hacia adelante en ambos strings y determina cuáles de esas letras pueden servir como punto de coincidencia, cada letra que esté presente en ambos substrings posibles abre una ruta de exploración independiente, la idea es avanzar en ambas cadenas hasta la primera aparición de la letra seleccionada y agregarla a la subsecuencia que se está construyendo. Una vez agregado ese carácter, el algoritmo repite el proceso desde las siguientes posiciones disponibles, buscando nuevas coincidencias hacia adelante.

Si en cierto punto no hay más letras comunes hacia adelante, esa ruta termina y la subsecuencia obtenida se compara con la mejor encontrada hasta el momento, el algoritmo repite todo este análisis comenzando desde distintas posiciones iniciales para asegurarse de cubrir todas las rutas posibles y finalmente la subsecuencia más larga de todas las rutas es la respuesta final.

Complejidad

La complejidad del algoritmo depende principalmente de que, desde cada punto de partida, se debe buscar letras comunes hacia adelante en ambas cadenas y abrir rutas independientes para cada coincidencia posible, pero en general como se van a recorrer ambos strings con posibles diferentes rutas, quedaría de la siguiente manera:

$$O(n \cdot m \cdot k)$$

Donde “n” sería la longitud del primer string, “m” la longitud del segundo y “k” las posibles rutas que fuera a tomar el algoritmo, esto a comparación al algoritmo del enunciado es bastante más eficiente

Paralelismo

La idea con paralelismo sería que cada hilo se encargue de empezar desde un punto de inicio diferente, en otras palabras, uno empieza en i en el string seleccionado, otro hilo en $i+1$, y así “n” veces hasta llegar al final, recorriendo el arreglo y comparando con el otro desde su punto de partida.