

Reflexión – Actividad integradora 1

Para encontrar partes del mcode dentro de nuestros archivos de transmisión utilizamos el algoritmo “kmp” además de la función prefix. En la función prefix se busca almacenar la distancia que ha de recorrer nuestro algoritmo kmp cuando no se haya alguna similitud con el mcode, esto sirve para que el kmp no haga comparaciones donde sabemos que no va a encontrar alguna similitud. El kmp va a comparar con el mcode, si no haya una coincidencia entonces recurre a nuestro vector auxiliar para “adelantar” el proceso, si encuentra una similitud entonces sigue con las comparaciones hasta que se complete, en ese momento quiere decir que encontramos un patrón e imprime dónde lo encontró. La complejidad de este proceso es lineal para ambos algoritmos.

Para la búsqueda de palíndromos empezamos con un loop for que irá desde el primer carácter de nuestro archivo hasta el último, luego con otro for que va desde el final del archivo (menos el valor de i en cada iteración) hasta encontrarse con i vamos revisando si en algunas de estas posiciones donde van iterando los loops tienen el mismo carácter, si sí significa que podría ser el inicio/final de nuestro palíndromo así que con el while vamos recorriendo desde estos extremos hasta el “centro” supuesto del palíndromo, si en algún punto de este proceso los caracteres son diferentes entonces continuamos con la búsqueda de caracteres iguales, si sí encontramos un palíndromo entonces comparamos su longitud para ver si encontramos el palíndromo más grande, si sí entonces se guarda la información de este. La complejidad de este proceso es de $O(n^2 \cdot m)$.

Por último, utilizamos el algoritmo “lcs” para encontrar el substring más largo entre nuestros dos archivos de transmisión. Creamos una tabla de 0 con longitud n (del tamaño de uno de los archivos de transmisión) y ancho m (tamaño del otro archivo), después se va comparando los caracteres de ambos archivos, si hay una coincidencia se le suma 1 a la coordenada superior izquierda de nuestra matriz, guardamos la coordenada con el valor más grande y el valor. Al final podemos sacar la posición final que está guardada en nuestro dato lcs, y la posición inicial restando el valor del substring a la posición final. La complejidad de este proceso es de $O(n \cdot m)$.