

Reflexión – Actividad integradora 1

Parte 1:

El algoritmo de la primera parte básicamente debe de encontrar todo el archivo de Mcode1, Mcode2 o Mcode3 y devolverte true si se encuentra el archivo o false en dado caso que no. Lo resolvimos con el algoritmo kmp, primero usamos la función prefix para conocer la tabla de prefijos-sufijos. Luego en la función kmp lo que hace es que es un ciclo que mientras no pase una variable del largo del archivo se va a seguir ejecutando luego checa si la posición actual del archivo transmisión coincide con el del mcode entonces se suma 1 a las 2 variables y así sigue hasta que el largo de las letras encontradas iguales al mcode lleguen al mismo largo del mcode en dado que llegue a eso entonces quiere decir que lo encontró y te imprime la posición inicial y final de donde se encuentra, en dado caso que no llegue a ser la siguiente letra igual la tabla de sufijos cambia a otro valor el cual ya tenía guardado con anterioridad, en dado caso que termine todo el documento y no encontró nada pasara a imprimir falso.

Su complejidad primero depende de la longitud del archivo y también de la longitud del mcode por lo que seria $O(m+n)$.

Parte 2:

El algoritmo dos básicamente pide que encuentre el palíndromo mas grande en los códigos transmisión, dicho código lo hicimos con 2 ciclos anidados, el primero se recorre desde 0 hasta el largo del código transmisión y el segundo va desde el largo del archivo hasta donde esta el primer ciclo, y mientras el extremo derecho y el izquierdo sean iguales se va a comparar una letra mas pegada al centro de cada extremo y si llega al centro y todas las letras fueron iguales de cada extremo entonces agrega ese palíndromo como el mas largo si después llega a encontrar otro mas grande entonces lo reemplaza, ese proceso se repite hasta que verifique todas las palabras posibles.

La complejidad seria $O(n^2*m)$ esto debido a que el primero es un ciclo que recorre todo el string, el segundo es uno que recorre todo el string menos i y eso también depende de m el cual es la comparación entre el extremo derecho y el izquierdo.

Parte 3:

El algoritmo debía de encontrar cadenas de caracteres que los 2 archivos tengan en común, y después de eso imprimirla. Para esto usamos el algoritmo llamado lcs el cual primero creamos una tabla de tamaño $(n+1)*(m+1)$. La función básicamente compara una carácter de 1 archivo con el del otro hasta encontrar cadenas en común, dichas cadenas va almacenando la mas grande junto con su posición, para así al final imprimir la posición inicial y final de la subcadenas de caracteres que tienen en común los 2 archivos de transmisión.

La complejidad es $O(n*n)$ ya que depende del largo del archivo transmisión 1 y del transmisión 2.

