

Actividad integradora 1

A00227534 - Marian Alejandra Herrera Ayala

En este trabajo, en conjunto con mi compañero, desarrollaremos un programa que en términos generales se encarga de verificar la existencia de la información almacenada en archivos mcodeX.txt en los archivos transmissionX.txt, carácter por carácter, así como la posibilidad de encontrar palíndromos de dentro de los últimos.

El programa fue conformado por tres partes, la primera encargada de verificar que los archivos mcode, efectivamente, se encuentren contenidos en alguno de los archivos transmission, en pocas palabras, estábamos trabajando con un algoritmo de búsqueda de patrones y, el nuestro, al final contó con una complejidad de $O(n)$, en el peor de los casos.

Mientras que, la tercera parte se utilizó un algoritmo para obtener el substring más largo que se tuvieran en común los archivos de transmission, la cual tuvo una complejidad de $O(n*m)$

En la segunda parte, tuvimos que desarrollar tres funciones distintas:

- **palindromeSearch(string input)** : Esta es la función de preparación para la búsqueda de palíndromos, el cual cuenta con una complejidad $O(n)$.
- **palindromeSearch(string input, int j, int k)** : Aquí se reciben dos valores, la posición inicial y la final de un string, durante esta función se estarán comparando la igualdad entre ambas posiciones y si en algún momento son diferentes, retornamos a la función de preparación. Por otro lado, si al final ambas posiciones son iguales, entonces llenamos los arreglos de longitudes con el tamaño del palíndromo y el vector posiciones se llena con las dos posiciones ingresadas. Esta función igual que la anterior cuenta con una complejidad de $O(n)$.
- **maxPalindrome()** : Esta función se ejecuta cuando los vectores longitudes y vectores estén llenos, y determina cual es el valor máximo almacenado en el vector longitud. Una vez obtenido el mencionado valor, se compara con los otros datos en el vector para ver si hay algún otro palíndromo que mida lo mismo, así en caso de ser así se imprime la información de todos los substring que tengan el tamaño máximo. En caso de no ser así, solo se imprime la información del palíndromo más largo. Esta función tiene una complejidad de $O(n)$.
- **impresionParte2(int p)** : Aquí solo se calculan en qué lugar del arreglo Posiciones se almacenaron las posiciones del palíndromo más largo, esta función presenta una complejidad de $O(1)$

Finalmente, el resto de las funciones se conformaron en funciones con complejidades que variaron entre $O(1)$ y $O(n)$.