Práctica 5. “Solución al problema de encontrara la subsecuencia común más larga mediante el uso de programación dinámica”

Benítez Morales Manuel Emilio Tellez Pérez Juan Manuel

I. Introducción

El problema de subsecuencia común más larga se trata de encontrar una subsecuencia común entre dos cadenas distintas de tal forma que esta sea de la mayor longitud posible.

Teniendo las cadenas:
X = {x1, x2, . . . , xn}
Y = {y1, y2, . . . , ym}
Z = {z1, z2, . . . , zq}

donde X e Y son la cadenas de entrada de longitud arbitraria y Z su subcadena común, el objetivo del algoritmo es obtener Z de la mayor longitud posible de manera eficiente, esto tiene diversa aplicaciones, por ejemplo:

Análisis de cadenas ADN: Estas cadenas pueden interpretarse en una computadora como secuencias de caracteres en el alfabeto, es posible analizar coincidencias y visualizar alteraciones de este, problemas de salud, parentesco familiar, etc.

Análisis de cambios en un archivo: El contenido de cualquier archivo es meramente una secuencia de caracteres, por lo cual es evidente cuando la secuencia cambia se pueden identificar estos cambios, un ejemplo de ello es el comando diff del bash de linux.

II. Implementación y pruebas

Realizado en lenguaje Python y basando en las diapositivas que se han proporcionado, se pudo crear este algoritmo.
Los pasos para este algoritmo son:

1. Crear dos matrices con las cuales guardaremos los estados con la longitud de la subsecuencia en cada estado y la otra será para guardar el camino a seguir a través de la matriz de estados.
2. Ahora recorreremos la matriz de estados pero teniendo en cuenta tres condiciones.
   La primera condición compara si los caracteres de las dos secuencias son iguales se suma uno más al estado anterior diagonalmente y guardamos la dirección “diagonal” en la otra matriz.
   La segunda condición compara los estados anteriores de la posición de arriba y de la izquierda. Si el estado de arriba es mayor entonces ese estado se replica en el actual y guardamos en la otra matriz la dirección “arriba”.
   La tercera condición es lo contrario, si el de la izquierda es mayor entonces lo guardamos en el estado actual y en la otra matriz guardamos la dirección “izquierda”.

3. Después de haber llenado las dos tablas, ahora recorreremos de manera recursiva la matriz donde guardamos las direcciones y así poder imprimir la subsecuencia más larga.

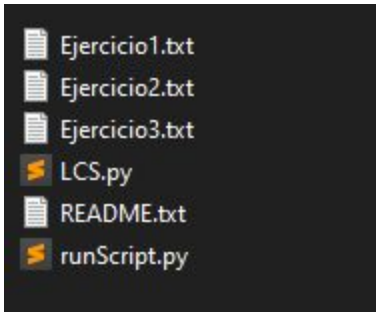

FIGURA 1. Archivos en directorio antes de la ejecución.

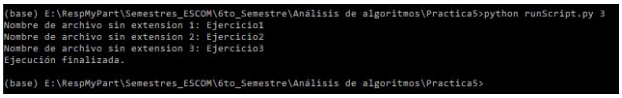

FIGURA 2. Ejecución del script.

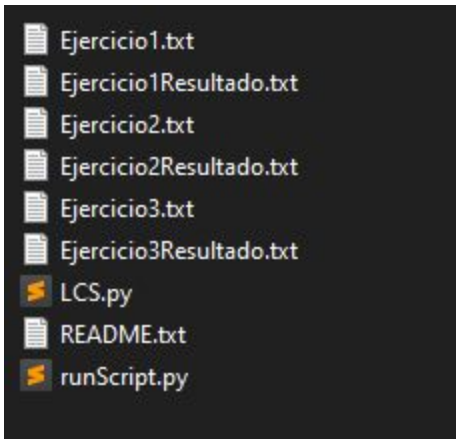

FIGURA 3. Archivos en directorio después de la ejecución.

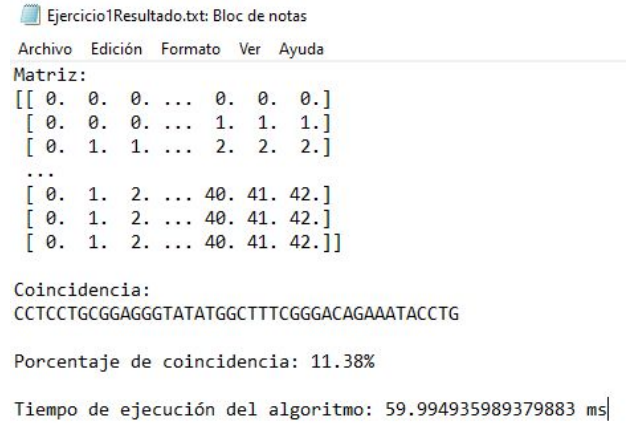

FIGURA 4. Ejemplo de contenido de los archivos resultantes.

III. Síntesis de "Of mice and men"

Las capacidades del cuerpo humano son especificadas por un programa único definido en una cadena de 3 millones de caracteres en el alfabeto {A, C, G, T} -Adenina, Citosina, Guanina, Timina- que conforman el ADN de cada persona.

El ADN puede verse como un programa estructurado que puede romperse en unidades más específicas , parecido a las subrutinas, estos son los genes.

Existen varias preguntas que surgen a partir del hecho de la computación como herramienta en el estudios de los genes, algunas son:

- Si se pueden encontrar coincidencias entre dos grandes cadenas de caracteres de ADN, provenientes del conocimiento que puede transferirse del conocimiento con animales pequeño, entonces el banco de genes conocidos crece, la pregunta es cómo se podría buscar en un diccionario así de grande utilizando la computación de manera eficiente.

- Como ya se ha mencionado, una cadena de ADN tiene millones de caracteres, sin embargo los métodos actuales pueden obtener solamente entre 500 y 700 caracteres de manera eficiente, entonces la pregunta queda en cómo se podrían ensamblar esas cadenas posibles de leer de manera coherente.
- Las secuencias que se obtienen suelen ser de varias especies, entonces la pregunta existente es si es posible encontrar una historia evolutiva de las especies a partir de analizar sus cadenas de ADN.

IV. Síntesis de "Biological applications"

Una de las aplicaciones del problema de LCS (Longest Common Subsequence) es el de comparar el ADN de distintos organismos. Entrando en contexto, una hebra de ADN consiste de una cadena de moléculas llamadas "bases", las "bases" posibles pueden ser adenina, guanina, citosina y timina. Éstas bases se representan por su letra inicial y se puede expresar una hebra de ADN como una cadena que se conforma de un conjunto finito de {A, C, G, T}.

Se pueden tener dos hebras que tengan una cadena distinta pero el propósito de compararlas es para saber qué tan similares son. Hay bastantes maneras de poder determinar esto, una de las formas podría ser si una cadena de ADN es una subcadena de otra. Otra forma sería averiguando qué tantos pasos se requieren para convertir una cadena de ADN a otra cadena si esta es más pequeña. Pero la que nos concierne realmente sería buscando una subcadena subsecuente común entre las dos cadenas de ADN que queremos comparar, y para saber qué tan similares son, debemos entrar la subcadena subsecuente más larga que tengan en común.

V. Conclusión

La programación dinámica en una de las herramientas más útiles y significativas dentro de la computación, otorgando la posibilidad de implementar este campo en otros más complejos para el beneficio del desarrollo.

Con esta práctica hemos practicado mucho más este estilo de programación y su aplicación para resolver tipos de problemas donde se tenga que utilizar, con este problema se pudo identificar varias aplicaciones donde la solución de este problema será una clave para la aplicación en varios ámbitos como por ejemplo en el área biológica donde se tiene que comparar cadenas de ADN y se quiere conocer qué tan similares son y la forma de resolverlo es con el resultado de este problema, el encontrar la subsecuencia más larga.