Universidad de Concepción
Departamento de Ing. Informática y Cs. de la Computación

Proyecto 3: Análisis de Algoritmos
Profes. Cecilia Hernández

Programación dinámica y Aproximados

La fecha de entrega es el Lunes 11 de Julio a las 23:59 horas.

1. Considere dos secuencias $X[1..n]$ y $Y[1..m]$ cuyo alfabeto está definido en el rango $aA..zZ$ para las secuencias de entrada. Sin embargo, las secuencias de salida pueden incluir el caracter $-$, que indica un espacio. Se desea maximizar su similitud la cual se mide mediante el cómputo de un puntaje que depende de un conjunto de condiciones. Este problema es importante en el área de bioinformática donde se desea encontrar la mejor coincidencia entre dos secuencias de ADN o aminoácidos. Las secuencias de ADN están formadas por los nucleótidos Adenina (A), Guanina (G), Citocina (C) y Timina (T); y las de aminoácidos, que representan proteínas, están formados por 20 caracteres (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y). Este problema en bioinformática se define como alineación de secuencias global y uno de los algoritmos mas conocidos utiliza programación dinámica conocido como el algoritmo de Needleman-Wunsch. (4 puntos)

   Para este proyecto considere que las condiciones para medir la similitud entre las dos secuencias son las siguientes.

   a) Si los caracteres de las dos secuencias coinciden se asigna un puntaje de 1.
   b) Si los caracteres no coinciden se asigna un puntaje -1.
   c) Si uno de caracteres en comparación es el caracter $-$ un puntaje de -2.

   Resuelva el problema usando programación dinámica y proporcione lo siguiente:

   a) Formulación mediante su recurrencia y la descripción de subproblemas a resolver.
   b) Solución top-down con memoización.

c) Solución bottom-up con tabulación.

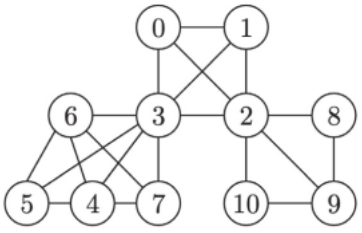
d) Complejidad algorítmica en tiempo y espacio.

Use la siguiente herramienta en línea para ver como se construye la tabla: `http://experiments.mostafa.io/public/needleman-wunsch/` Por ejemplo, defina $X = \{CCAATTAG\}$ e $Y = \{CAATTTG\}$ donde se obtienen las secuencias de salida $\{CCAATTAG\}$ y $\{C-AATTTG\}$ con puntaje 3.

2. Para el problema de mínimo vertex cover. (2 puntos)

a) Implemente el algoritmo 2-aproximado visto en clases. Proporcione un ejemplo donde el algoritmo aproximado obtiene el valor óptimo y uno donde obtiene el doble del óptimo.

b) Suponga un problema donde le proporcionan los datos asociados a un grafo, pero en lugar de entregarle los vértices con sus listas de adyancencia o las aristas, solo le proporcionan los cliques maximales que cubren el grafo. Construya un grafo de cliques y analice si es posible encontrar el número mínimo de cliques que cubren el grafo de cliques. Vea si puede aplicar de alguna manera el algoritmo 2-aproximado de mínimo vertex cover para este problema o no, o bien si podría definir restricciones para permitirlo. La siguiente figura proporciona un ejemplo con la lista de cliques maximales que cubren el grafo dado, donde C_i es un clique maximal y está definido por los vértices que lo definen. Recuerde que un clique maximal es un clique que no puede crecer, es decir no puede contener un vértice adicional tal que existan todas las aristas en el grafo.



$C_0 : 0, 1, 2, 3$
$C_1 : 3, 4, 6, 7$
$C_2 : 3, 4, 5, 6$
$C_3 : 2, 8, 9$
$C_4 : 2, 9, 10$

Observación

La entrega consiste en un informe de no más de 5 páginas, código fuente (en C o C++) documentado y un readme.txt que describa brevemente que hace su implementación, como compilar y ejecutar incluyendo un ejemplo.