

UNIVERSIDAD DE GRANADA
E.T.S. DE INGENIERÍAS INFORMÁTICA
y DE TELECOMUNICACIÓN



Departamento de Ciencias de la
Computación e Inteligencia Artificial

Algorítmica

Guión de Prácticas

Práctica 4: Programación Dinámica

Curso 2019-2020

Grado en Informática

1. Normativa

El objetivo de esta práctica es que el estudiante sea capaz de analizar un problema y plantear una solución al mismo mediante la técnica de Programación Dinámica. Para ello, se propone un ejercicio guiado y un ejercicio común que deberá ser resuelto por cada equipo.

Cada equipo tendrá que entregar:

1. Una memoria en la que se describa la solución propuesta para el problema común.
2. En la memoria se debe añadir el pseudocódigo del algoritmo propuesto, y varios escenarios de ejecución.
3. El código elaborado para resolver el problema con su correspondiente *Makefile*.

2. Ejercicio guiado: Subsecuencia de caracteres más larga

Este ejercicio se encuentra resuelto en <https://algorithms.tutorialhorizon.com/dynamic-programming-longest-common-subsequence>. El alumno deberá estudiar la metodología de diseño seguida, para realizar posteriormente el diseño e implementación de un algoritmo que resuelva el problema del apartado 3 de este documento.

Sean dos secuencias de caracteres $X = (x_1, x_2, \dots, x_m)$ e $Y = (y_1, y_2, \dots, y_n)$, de longitudes m y n respectivamente. El problema consiste en encontrar la máxima subsecuencia de caracteres común que aparecen en ambas cadenas de izquierda a derecha (no necesariamente de forma contigua). Por ejemplo, para las cadenas $S = \text{ABAZDC}$ y $T = \text{BACBAD}$, la máxima subsecuencia común tiene longitud 4 y es ABAD , siendo localizadas en $S = \text{ABAZDC}$ y en $T = \text{BACBAD}$ (ver Figura 1).

String A	a	c	b	a	e	d
String B	a	b	c	a	d	f

Figura 1: Ejemplo de subsecuencia de caracteres más larga.

3. Problema del Viajante de comercio

En su formulación más sencilla, el problema del viajante de comercio (TSP, por sus siglas en inglés *Traveling Salesman Problem*) se define como sigue: dado un conjunto de ciudades y una matriz con las distancias entre todas ellas, un viajante debe recorrer todas las ciudades exactamente una vez, regresando al punto de partida, de forma tal que la distancia recorrida sea mínima. Más formalmente, dado un grafo G , conexo y ponderado, se trata de hallar el ciclo Hamiltoniano de mínimo peso de ese grafo. Las pruebas del algoritmo pueden realizarse con los mismos datos de prueba empleados en la práctica 3, usando además la distancia euclídea como criterio de evaluación. De igual modo, la visualización de las soluciones pueden realizarse usando *gnuplot*, al igual que en la práctica 3.

Tareas a realizar

1. Diseñar e implementar un algoritmo basado en Programación Dinámica para el problema del Viajante del Comercio.
2. Comparar la solución y eficiencia del algoritmo propuesto por Programación Dinámica con los elaborados siguiendo una estrategia voraz.