ANÁLISIS Y DISEÑO DE ALGORITMOS I

PRÁCTICO Nº 7. PROGRAMACIÓN DINÁMICA

Construya algoritmos por la técnica Programación Dinámica para los siguientes problemas. Determine para cada uno su **complejidad temporal**.

1. **Problema de la Mochila**. Se tienen n objetos y una mochila. Para i = 1, 2,..n, el objeto i tiene un peso positivo p_i y un valor positivo v_i. La mochila puede llevar un peso que no sobrepase P. Los objetos no pueden ser fraccionados. El objetivo es llenar la mochila de manera tal que se maximice el valor de los objetos transportados, respetando la limitación de capacidad impuesta.

2. **Cambio de monedas**. Dado un conjunto C de N tipos de monedas con un número ilimitado de ejemplares de cada tipo, se requiere formar, si se puede, una cantidad M empleando el mínimo número de ellas. Por ejemplo, un cajero automático dispone de billetes de distintos valores: 100$, 25$, 10$, 5$ y 1$, si se tiene que pagar 289$, la mejor solución consiste en dar 10 billetes: 2 de 100$, 3 de 25$, 1 de 10$ y 4 de 1$.

3. **Triangulación de polígonos**. Dados los vértices de un polígono y la distancia entre cada par de ellos, se requiere seleccionar un conjunto de cuerdas (líneas entre los vértices no adyacentes) que no se corten y que dividan al polígono en triángulos con un costo mínimo (definimos al costo como la suma de las distancias asociadas a las cuerdas).

4. **Problema del Viajante**. Un viajante de comercio debe visitar una serie de ciudades. Cada ciudad está conectada con las restantes mediante carreteras de longitud conocida. El problema consiste en hallar la ruta que deberá tomar para visitar todas las ciudades retornando a la ciudad de partida.

5. **Caminos mínimos**. Dado un conjunto de ciudades de una región y la distancia entre cada par de ciudades vecinas (ciudades unidas por una ruta directa sin ciudades intermedias), encontrar el camino mínimo entre cada par de ellas.

6. **Multiplicación de matrices**. Sea $M = M_1 \text{ x } M_2 \text{ x } M_3 \text{ x } M_4$ una multiplicación encadenada de matrices. La multiplicación es asociativa, por lo tanto hay varias maneras de resolver secuencias de productos matriciales. Resuelva este problema utilizando programación dinámica.

7. **Problema de la subsecuencia más larga**. Dada una secuencia X={x1 x2 ... xm}, decimos que Z={z1 z2 ... zk} es una subsecuencia de X (siendo k ≤ m) si existe una secuencia creciente {i1 i2 ... ik} de índices de X tales que para todo j = 1, 2, ..., k tenemos xij = zj.
   Ej: Z={BCDB} es una subsecuencia de X={ABCBDAB}con la correspondiente secuencia de índices {2,3,5,7}.
   Dadas dos secuencias X e Y, decimos que Z es una subsecuencia común de X e Y si es subsecuencia de X y subsecuencia de Y.
   Implemente un algoritmo para determinar la subsecuencia de longitud máxima común a dos secuencias.

8. **Gramática**. Dada una gramática $G = < N, T, P, S >$:
   - N: alfabeto de no terminales
   - T: alfabeto de terminales
   - P: conjunto de reglas de producción
   - S: elemento distinguido

   Construya un programa para determinar si una cadena $w = a_1a_2...a_n$ pertenece al lenguaje.