

Algoritmos y Estructuras de Datos II - 1º cuatrimestre 2019
Práctico 3 - Parte 3: Programación dinámica.

1. Dar una definición de la función cambio utilizando la técnica de programación dinámica a partir de la siguiente definición recursiva (backtracking):

$$cambio(i, j) = \begin{cases} 0 & j = 0 \\ \infty & j > 0 \wedge i = 0 \\ \min_{q \in \{0, 1, \dots, j \div d_i\}} (q + cambio(i - 1, j - q * d_i)) & j > 0 \wedge i > 0 \end{cases}$$

2. Para el ejercicio anterior, ¿es posible completar la tabla de valores “de abajo hacia arriba”? ¿Y “de derecha a izquierda”? En caso afirmativo, reescribir el programa. En caso negativo, justificar.
3. Dar una definición de la función cambio utilizando la técnica de programación dinámica a partir de cada una de las siguientes definiciones recursivas (backtracking):

(a)

$$cambio(i, j) = \begin{cases} 0 & j = 0 \\ 1 + \min_{i' \in \{1, 2, \dots, i \mid d_{i'} \leq j\}} (cambio(i', j - d_{i'})) & j > 0 \end{cases}$$

(b)

$$cambio(i, j) = \begin{cases} 0 & j = 0 \\ \infty & j > 0 \wedge i = n \\ cambio(i + 1, j) & d_i > j > 0 \wedge i < n \\ \min(cambio(i + 1, j), 1 + cambio(i, j - d_i)) & j \geq d_i > 0 \wedge i < n \end{cases}$$

4. Ejecutar paso a paso el algoritmo de Floyd para el siguiente grafo

$$\begin{array}{llll} w((1, 2)) = 7 & w((2, 3)) = 4 & w((3, 6)) = 4 & w((5, 6)) = 6 \\ w((1, 6)) = 3 & w((2, 4)) = 2 & w((3, 8)) = 6 & w((6, 7)) = 5 \\ w((1, 7)) = 5 & w((2, 5)) = 1 & w((4, 6)) = 8 & w((8, 5)) = 2 \\ w((1, 3)) = 3 & w((3, 4)) = 5 & w((5, 4)) = 3 & w((8, 7)) = 3 \end{array}$$

5. Para cada una de las soluciones que propuso a los ejercicios del 2 al 8 del práctico de backtracking, dar una definición alternativa que utilice la técnica de programación dinámica. En los casos de los ejercicios 2, 4 y 5 modificar luego el algoritmo para que no sólo calcule el valor óptimo sino que devuelva la solución que tiene dicho valor (por ejemplo, en el caso del ejercicio 2, cuáles serían los pedidos que debería atenderse para alcanzar el máximo valor).