

Algoritmos y Estructuras de Datos II, Segundo del Grado de Ingeniería Informática, Test de Programación Dinámica, mayo 2016

Para los dos problemas siguientes, explicar cómo se pueden resolver el problema por Programación Dinámica, indicando:

- El número elementos en la solución final y los subproblemas que se consideran para llegar a esa solución.
- La ecuación de recurrencia que relaciona la solución para un tamaño en función de tamaños menores, y cuáles son y los valores de los casos base.
- Las estructuras de datos (tablas, vectores...) a utilizar para guardar las subsoluciones y cómo se rellenan estas estructuras.
- Dónde se encuentra la solución final o cómo se recompone a partir de la información de las tablas.
- Una estimación (cuanto más precisa mejor) del tiempo de ejecución del método de resolución del problema por Programación Dinámica.
- Explicar cómo funciona sobre el ejemplo que se proporciona.
- Si la solución sigue un esquema similar a algún otro de los ejemplos vistos en clase de teoría o prácticas, o a los que aparecen en el texto guía de la asignatura u otra bibliografía, indicar claramente el problema o donde se encuentra en la bibliografía, y las diferencias de la solución aportada con respecto a la del problema al que se asemeja.

1) Queremos comprar algunos productos de entre una serie de n productos. Cada producto i tiene un coste c_i , un beneficio b_i y se puede dividir en p_i partes. Disponemos para gastarnos una cantidad C . Queremos comprar productos o partes de productos maximizando el beneficio. Si de un producto con coste c , beneficio b y que se puede dividir en p partes compramos q partes (con $0 \leq q \leq p$) el beneficio que se obtiene es $\frac{q}{p}b$ y lo que se gasta en él es $\frac{q}{p}c$.

Mostrar el funcionamiento con $n = 4$, $b = (3, 2, 4, 3)$, $c = (3, 1, 3, 2)$, $p = (2, 3, 1, 2)$, $C = 7$.

2) Tenemos un damero cuadrado, con algunas posiciones marcadas con D o A. Se quiere obtener el camino de longitud mínima para llegar desde la posición superior izquierda (casilla $(1, 1)$) a la inferior derecha (casilla (n, n)). Desde una casilla se puede avanzar a una casilla vecina a la derecha o abajo. En el tablero hay casillas marcadas con D y con A. Cuando al avanzar desde una casilla se llega a una casilla etiquetada con D, se avanza a la casilla a la derecha a la etiquetada con D; si al avanzar se llega a una etiquetada con A se avanza una casilla abajo. Si en un avance por propulsión desde una casilla con D o A se llega a otra también etiquetada, se sigue avanzando. La longitud de un camino es el número de pasos de que consta, y no se contabilizan como pasos las propulsiones desde casillas etiquetadas.

El ejemplo a utilizar es

	D	D	
		A	
D	A	D	
		A	