

Problemas Tema 4: Programación dinámica

Miguel Ángel Guerrero y Jorge Guillamón

11 de junio de 2020

Resumen

Resolución de los problemas 4 y 7 del Tema 4

Problema 4

Alí Babá ha conseguido entrar en la cueva de los ciento un mil ladrones, y ha llevado consigo su camello junto con dos grandes alforjas; el problema es que se encuentra con tanto tesoro que no sabe ni qué llevarse. Los tesoros son joyas talladas, obras de arte, cerámica... es decir, son objetos únicos que no pueden partirse ya que entonces su valor se reduciría a cero. Afortunadamente los ladrones tienen todo muy bien organizado y se encuentra con una lista de todos los tesoros que hay en la cueva, donde se refleja el peso de cada pieza y su valor en el mercado de Damasco. Por su parte, Alí sabe la capacidad de peso que tiene cada una de las alforjas. Diseñar un algoritmo que, teniendo como datos los pesos y valor de las piezas, y la capacidad de las dos alforjas, permita obtener el máximo beneficio que podrá sacar Alí Babá de la cueva de las maravillas.

La cuestión se ha afrontado, en esencia, resolviendo el problema de la mochila (punto 3 de la teoría) dos veces seguidas. Esta resolución ha presentado dos modificaciones con respecto al problema original:

- Se tienen que actualizar los objetos tras llenar la primera bolsa, se entiende que un objeto no puede estar en dos bolsas distintas.
- Se tiene que prestar atención al orden en que se llenan las alforjas, qué alforja llenamos primero es relevante para el problema.

Resolución

Se almacenan los objetos en una lista con elementos [peso, valor].
Se hace una copia de la lista y se resuelve el problema para un orden de alforjas.
Se resuelve el problema para el orden de alforjas inverso.
Se devuelve la opción con más valor en forma de dos listas con elementos [peso, valor]

función ladrón

Se encarga de resolver el problema de la mochila clásico.

Devuelve: los mejores objetos a llevar en una bolsa

Entrada: p1: capacidad bolsa; elementos: Lista de [peso_i, valor_i]

Salida: lista con [peso_i, valor_i]

La lista de la entrada se ve modificada en la salida.

El programa está listo para ser ejecutado.

Problema 7

Se desea determinar la subsecuencia común de dos secuencias de 0s y 1s A y B que tenga la longitud máxima, para lo que se pide:

- Explicar con detalle la forma de resolver el problema.
- Hacer un algoritmo de Programación Dinámica que obtenga la longitud máxima posible y una secuencia común de dicha longitud.

Explicación

Toda secuencia de $n+1$ caracteres es también una subsecuencia de n caracteres. Gracias a esto es aplicable el principio de optimalidad y por tanto el problema admite una solución mediante programación dinámica.

La solución que hemos encontrado pasa por ir cribando una lista, inicialmente llena con las subcadenas coincidentes de longitud 1, a medida que se aumenta el tamaño. La solución será finalmente una de las últimas (o la última cadena) que quede en la lista antes de que se vacíe por completo.

Las ocurrencias de subcadenas se representan en memoria por medio de la tripleta `[s1_i, s2_i, size]` siendo los dos primeros índices donde comienza cada subcadena y `size` su tamaño.

Para encontrar las subcadenas de longitud 1 más básicas se ha usado el producto cartesiano de las ubicaciones de los 1s y 0s. Ejemplo: cadenas '101' y '110', los índices del 1 son para cada una `[0,2]` y `[0,1]`, el producto cartesiano de estas listas da como resultado: `[[0,0], [0,1], [2,0], [2,1]]`.

Programa

Puede encontrar el código comentado junto a esta memoria.