

# Taller 3: Programación Dinámica

Universidad Nacional de Colombia, Bogotá

Resuelva los problemas listados más abajo usando programación dinámica. Para cada uno de estos problemas, debe realizar lo siguiente:

- (a) Muestre que el problema cumple con la propiedad de subestructura óptima.
- (b) Plantee una ecuación recursiva para resolver el problema. Debe explicar claramente la ecuación, todas las variables que esta usa y su relación con el problema.
- (c) Escriba en pseudocódigo un algoritmo *top-down with memoization* que resuelva el problema. No solo se debe encontrar el valor óptimo, sino la solución que lo produce.
- (d) Escriba en pseudocódigo un algoritmo de programación dinámica *bottom-up* que resuelva el problema. No solo se debe encontrar el valor óptimo, sino la solución que lo produce.
- (e) Implemente los dos algoritmos de los items anteriores en Python de manera que resuelva el problema de manera eficiente. (Debe mostrar el código del programa en el PDF y anexarlo en su extensión de Python).
- (f) Ilustre de manera gráfica un ejemplo de tamaño 10 donde se muestre que hay varios subproblemas y la solución encontrada. Muestre las tablas de solución de todos los subproblemas resueltos.

Los problemas que debe resolver siguiendo los pasos anteriores son:

1. **Problema de la Mochila de Enteros (con ítems duplicados).** Se tienen  $n$  tipos de ítems, donde el  $i$ -ésimo elemento tiene tamaño entero  $s_i$  y costo real  $v_i$ . Se trata de llenar la mochila de capacidad  $C$  con una selección de ítems que maximicen el costo total. Se pueden agregar múltiples ítems del mismo tipo en la mochila.
2. **Problema de la Mochila de Enteros (sin ítems repetidos).** Se tienen  $n$  tipos de ítems, donde el  $i$ -ésimo elemento tiene tamaño entero  $s_i$  y costo real  $v_i$ . Se trata de llenar la mochila de capacidad  $C$  con una selección de ítems que maximicen el costo total. NO se pueden agregar múltiples ítems del mismo tipo en la mochila.
3. **Subsecuencia Contigua de Valor Máximo** Resuelva el problema de hallar la subsecuencia contigua  $A_i$  a  $A_j$ , de una secuencia de reales  $A_1, A_2, \dots, A_n$ , cuya suma sea máxima.
4. **Subsecuencia Creciente más Larga** Resuelva el problema de hallar la subsecuencia creciente más larga de una secuencia de números reales  $A_1, A_2, \dots, A_n$ . Recordar que los elementos no tienen que ser consecutivos necesariamente.
5. **Distancia de Edición.** Resuelva el problema de hallar la distancia de edición entre dos cadenas de texto. Dadas dos cadenas de texto  $A$  de longitud  $n$  y la cadena de texto  $B$  de longitud  $m$ , se requiere transformar la cadena  $A$  en la cadena  $B$  con el mínimo número de operaciones posibles. Las operaciones disponibles son: borrar un caracter de  $A$ , insertar un caracter en  $A$ , y reemplazar un caracter de  $A$  por otro caracter. El mínimo número de operaciones para transformar  $A$  en  $B$  es la distancia de edición entre  $A$  y  $B$ .
6. **Cajas Apiladas.** Resuelva el siguiente problema: Dadas  $n$  cajas, donde la  $i$ -ésima caja tiene dimensiones  $h_i \times w_i \times d_i$ , encontrar la forma de apilar las cajas de manera que se maximice la altura de la pila. La restricción es que sólo se puede poner una caja encima de otra si las dimensiones 2D de la base de la caja de abajo son estrictamente más grande que las dimensiones de la base 2D de la caja de arriba.

Bibliografía: Brian Dean, 2002.