# Práctico 5 ALGABO: Ejercicios de programación dinámica.

Mauricio Velasco

1. **Alineación de secuencias.** Ejecute a mano el método de programación dinámica para encontrar la mejor alineación entre las secuencias $GATTACA$ y $TAGGACA$ asumiendo que un gap tiene costo $+1$ y $\alpha_{xy} = +3$ para $x \neq y$. Calcule la penalidad mínima y exhiba una alineación que alcance ese mínimo. Justifique claramente todos sus pasos.

2. **Como cortar un tubo?** Tenemos longitudes (enteras positivas) $\ell_1 < \ldots < \ell_k$ deseadas y precios fijos (reales positivos) $p_1, \ldots, p_k$ para esas longitudes. Queremos cortar un tubo de longitud $L$ (entera) en piezas de las longitudes $\ell_i$ de tal forma que se maximice la ganancia.

   *a*) Formule la pregunta como un problema de optimización (Queremos determinar el número $x_i$ de piezas de longitud $\ell_i$ a ser cortadas).

   *b*) Proponga un algoritmo de programación dinámica para resolver este problema. Formule y demuestre su ecuación de Bellman.

   *c*) Describa un espacio de estados para el problema.

   *d*) Implemente su algoritmo y aplíquelo para resolver el problema con longitudes $[1, 2, 3, 4, 5, 6]$ y precios respectivos $[1, 5, 9, 12, 12, 15]$ para un tubo de longitud 10.

3. **Árboles de búsqueda.** Considere la siguiente tabla de números y probabilidades respectivas

| valor | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| probabilidad | 0.5 | 0.25 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |

   implemente un algoritmo de programación dinámica para encontrar el árbol de búsqueda óptimo para estos números.

   *a*) Dibuje el árbol óptimo resultante

b) Calcule el tiempo esperado de búsqueda para su árbol óptimo.

c) Dibuje el árbol binario lleno con items $[7, 6, 5, 4, 3, 2, 1]$ (usando la notación de la clase sobre heaps) y calcule el tiempo esperado de búsqueda para ese árbol con las probabilidades de arriba.

4. Dadas matrices $A_1, \ldots, A_n$ de tamaños $a_i \times a_{i+1}$ para $i = 1, \ldots, n$ queremos calcular su producto $A_1 \ldots A_n$.

   a) Demuestre las siguientes afirmaciones:

      1) El resultado del producto no depende de cómo asociemos las matrices es decir $(A_1A_2)A_3 = A_1(A_2A_3)$
      2) El número de operaciones necesarias (sumas y multiplicaciones) para calcular el producto $A_1A_2$ mediante la fórmula usual es $O(a_1a_2a_3)$.

   b) Aunque el resultado no depende de como asociamos la matrices, el numero de operaciones realizadas SI depende de como las asociamos. Encuentre tres matrices concretas $A_1, A_2$ y $A_3$ donde el cálculo de $(A_1A_2)A_3$ requiera mil veces el número de operaciones usadas para calcular $A_1(A_2A_3)$.

   c) Demuestre que hay una correspondencia entre maneras de asociar (poner paréntesis) el producto de nuestras $n$ matrices y árboles binarios con $n$ hojas. Escriba todas las maneras de asociar $n = 4$ matrices con sus árboles binarios correspondientes.

5. Proponga un algoritmo usando programación dinámica para encontrar la manera de asociar más eficiente (la que usa el mínimo número de operaciones) para calcular el producto $A_1 \ldots A_n$ de las matrices del ejercicio anterior.

   a) Escriba la ecuación de Bellman de su problema.

   b) Describa el espacio de estados.

   c) Estime el tiempo de ejecución del algoritmo.