UNIVERSIDAD DE MÁLAGA	**Análisis y Diseño de Algoritmos** Práctica 3

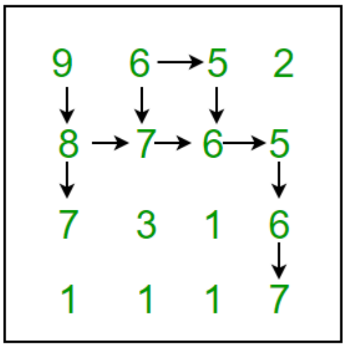
1.- Sean $A_1,\ldots,A_n$ un conjunto de matrices que se desean multiplicar en secuencia. Se supone que estas matrices son compatibles, esto es, la dimensión de A_i es $m_i \times n_i$ y se cumple que $n_i = m_{i+1}$. Nótese que al multiplicar dos matrices de tamaños $m{\times}n$ y $n{\times}p$ se realizan $m{\cdot}n{\cdot}p$ multiplicaciones. El número total de multiplicaciones escalares que se realizan dependerá del modo en que se asocien las matrices para la multiplicación. Por ejemplo, sea $A_1^{10\times100}$, $A_2^{100\times5}$ y $A_3^{5\times50}$:

- $((A_1A_2)A_3)$ supone $10 \cdot 100 \cdot 5 + 10 \cdot 5 \cdot 50 = 7500$ multiplicaciones escalares.
- $(A_1(A_2A_3))$ supone $100 \cdot 5 \cdot 50 + 10 \cdot 100 \cdot 50 = 75000$ multiplicaciones escalares.

Resulta por lo tanto fundamental encontrar la asociación óptima para realizar la multiplicación.

a) Habiéndose caracterizado el problema M_{ij} como el número óptimo de multiplicaciones escalares necesarias para multiplicar $A_iA_{i+1} \cdots A_j$, encontrar una expresión recurrente para M_{ij}.
b) Implementar un algoritmo de programación dinámica bottom-up que se base en la recurrencia 1b y calcule el número óptimo de multiplicaciones escalares necesarias para multiplicar una serie de matrices $A_1,\ldots,A_n$.
c) Modificar el algoritmo 1b para mostrar el contenido de la tabla auxiliar una vez que está rellena.
d) Implementar una versión memoizada del algoritmo recursivo simple que emana de la recurrencia definida en 1b.

2.- Una secuencia serpentine está formada por números adyacentes en la matriz, donde los número siguientes en la secuencia al número a_{ij} deben verificar $a_{i,j+1} = a_{ij} \pm 1 \vee a_{i+1,j} = a_{ij} \pm 1$, es decir, la secuencia podría continuar con el número a su derecha o justo debajo si tienen el valor a_{ij} *+1* ó *-1*.
Por ejemplo, en la matriz dada las posibles secuencias serpentinas son (9,8,7), (9,8,7,6,5,6,7), (6,7,6,5,6,7) y (6,5,6,5,6,7). Entre ellas la de máxima longitud es (9, 8, 7, 6, 5, 6, 7)



a) Diseñar una ecuación recursiva que calcule la longitud de la secuencia serpentina de máxima longitud que se puede encontrar en una matriz dada si queremos que dicha secuencia acabe en la posición (i,j).
b) Implementar un algoritmo de programación dinámica que devuelva la secuencia de máxima longitud que acaba en la posición (i,j) dada.