Examen de Teoría de Algoritmos

Cuarto Curso de la Licenciatura de Informática. 18 de septiembre de 1997

1. Considérese el siguiente problema:

   Tenemos un conjunto de n eslabones $(e_1, e_2, \ldots, e_n)$ que nos permitirán componer una cadena (no cerrada). Para formar la cadena, tenemos que unir cada eslabón e_i con el e_{i+1}, $i = 1, 2, \ldots, n-1$. Asociado a cada eslabón, tenemos el peso del mismo $(p_1, p_2, \ldots, p_n)$. Los distintos eslabones se irán agrupando sucesivamente en subcadenas. El coste de agrupar dos subcadenas c_i y c_j, $COSTO(c_i \cup c_j)$, es igual a la suma de los pesos de los eslabones en los extremos de las mismas. Por ejemplo, suponer las cadenas $c_1 = e_i - e_{i+1} - \ldots - e_{k-1} - e_k$; $c2 = e_{k+1} - e_{k+2} - \ldots - e_m$ y $c_3 = e_{m+1}$. En este caso, $COSTO(c_1 \cup c_2) = p_i + p_k + p_{k+1} + p_m$ y $COSTO(c_2 \cup c_3) = p_{k+1} + p_m + p_{m+1}$. Para unir los n eslabones de la cadena, serán necesarias $n-1$ uniones, definiendo el $COSTO-TOTAL$ para formar la cadena como la suma de los costos necesarios en cada una de estas uniones.

   Utilizando la técnica de Programación Dinámica, diseñar un algoritmo que nos permita resolver el problema anterior con un menor $COSTO - TOTAL$ (**3 puntos**).

2. Resolver el problema de la pregunta anterior utilizando la técnica Backtracking (**2 puntos**).

   Nota: Para resolver el problema, primero dar la representación del árbol del espacio de estados y posteriormente dar el algoritmo.

3. Sea $T(n)$ la ecuación de recurrencia de un algoritmo *Divide y Vencerás*. Si $T(n)$ es de la forma $T(n) = aT(n/b) + n$, con $a \geq 1, b > 1$, analizar el tiempo de ejecución del algoritmo dependiendo de los valores de a y b (**1'5 puntos**).

4. Resolver exactamente la siguiente recurrencia, siendo n una potencia de 2:

$$T(n) = \begin{cases} 1 & \text{si } n = 1 \\ 2T(n/2) + \lg n & \text{en caso contrario} \end{cases}$$

   Exprese su respuesta en el modo más sencillo posible empleando la notación θ (**1 punto**).

5. Diseñar un algoritmo *branch and bound* para resolver el problema de asignación que tiene asociada la siguiente matriz (**2.5 puntos**):

	1	2	3	4	5
a	11	17	8	16	20
b	9	7	12	6	15
c	13	16	15	12	16
d	21	10	12	11	15
e	14	10	12	11	15