Teoría de Algoritmos
Segundo de Ingeniería Informática
Examen de Septiembre del Curso 2003-2004

1. Demostrar la veracidad o falsedad de las siguientes propiedades:
   a) Regla de la suma: Si f_1 es $\Omega(g)$ y f_2 es $\Omega(h)$ entonces $f_1 + f_2$ es $\Omega(g + h)$
   b) Regla del producto: Si f_1 es $\Omega(g)$ y f_2 es $\Omega(h)$ entonces $f_1 \cdot f_2$ es $\Omega(g \cdot h)$
   c) Si existe

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = k.$$

   dependiendo de los valores que tome k obtenemos:
   i) Si $k \neq 0$ y $k < \infty$ entonces $\Omega(f) = \Omega(g)$
   ii) Si $k = 0$ entonces g es $\Omega(f)$, es decir, $\Omega(g) \subset \Omega(f)$, pero f no es $\Omega(g)$

2. Una de las cuestiones a considerar cuando se diseña un algoritmo mediante la técnica Divide y Vencerás es la partición y el reparto equilibrado de los subproblemas. Más concretamente, en el problema de la búsqueda binaria nos podemos plantear las dos siguientes cuestiones:
a) supongamos que en vez de dividir el vector de elementos en dos mitades del mismo tamaño, las dividimos en dos partes de tamaños 1/3 y 2/3. ¿Conseguiremos de esta forma un algoritmo mejor que el original?.
b) podemos plantearnos también diseñar un algoritmo de búsqueda "ternaria", que primero compare con el elemento en posición $n/3$ del vector, si éste es menor que el elemento x a buscar entonces compare con el elemento en posición $2n/3$, y si no coincide con x busque recursivamente en el correspondiente subvector de tamaño 1/3 del original. ¿Conseguiremos así un algoritmo mejor que el de búsqueda binaria?

3. Un informático necesita diseñar n programas urgentemente, y sabe de antemano el tiempo que le va a llevar el diseño de cada uno de ellos: en el programa i-ésimo tardará t_i minutos. Como en su empresa le pagan dependiendo de la satisfacción del cliente, necesita decidir el orden en el que se pondrá a preparar los programas para minimizar el tiempo medio de espera de los clientes. En otras palabras, si llamamos E_i a lo que espera el cliente i-ésimo hasta disponer de su programa, necesita minimizar la expresión:

$$E(n) = \sum_{i=1}^{n} E_i \ .$$

Deseamos comprobar si este problema puede resolverse con un algoritmo greedy. Si así fuera, queremos diseñar un algoritmo de ese tipo que resuelva el problema y probar su validez.

4. Sea G = (X, A) un grafo dirigido con un conjunto X de n vértices y otro A de arcos. Diseñar un algoritmo que permita conocer si dos vértices de un grafo están conectados o no. Comprobar si el problema puede resolverse con Programación Dinámica, y si ese es el caso, calcular la eficiencia del correspondiente algoritmo.

5.Definir que se entiende por restricción implícita y explicita, en general, y concretar las definiciones en el caso del Problema del Movimiento del Rey de Ajedrez, que consiste en lo siguiente: Dado un tablero de ajedrez de tamaño nxn, se coloca un rey en una casilla arbitraria de coordenadas (x,y). El problema consiste en determinar los $n^2 - 1$ movimientos del rey de forma que todas las casillas del tablero sean visitadas una sola vez, si tal secuencia de movimientos existe.

- **Tiempo para la realización del examen: 3 horas**
- **No está permitido el uso de apuntes, libros o cualquier otro material de consulta**