

## Examen de Teoría de Algoritmos.

Curso 2010–2011. Convocatoria extraordinaria de septiembre

### I.T.I. Gestión e I.T.I. Sistemas

1. (2 pt.) Sea  $T(n)$  la ecuación de recurrencia de un algoritmo Divide y Vencerás. Si  $T(n)$  es de la forma  $T(n) = aT(n/b) + n^k$  con  $a \geq 1$ ,  $b > 1$ , resolver dicha ecuación y analizar el orden de ejecución del algoritmo dependiendo de los valores de  $a$  y  $b$ .
2. (1 pt.) Utilizar la técnica Divide y Vencerás para diseñar un algoritmo que permita calcular el máximo y el mínimo de un vector de forma eficiente.
3. (2 pt.) Sean  $n$  programas  $P_1, P_2, \dots, P_n$  que hay que almacenar en un disco. El programa  $P_i$  requiere  $s_i$  KiloBytes de espacio y la capacidad del disco es  $D$  KiloBytes, donde  $D < \sum_{i=1}^n s_i$ . Resolver las siguientes cuestiones:
  - a) Se desea maximizar el número de programas almacenados en el disco. Demostrar que podemos utilizar un algoritmo *greedy* que seleccione los programas por orden no decreciente de  $s_i$  para obtener la solución exacta o dar un contraejemplo en caso contrario.
  - b) Se desea utilizar la mayor capacidad posible del disco. Demostrar que podemos utilizar un algoritmo *greedy* que seleccione los programas por orden no decreciente de  $s_i$  para obtener la solución exacta o dar un contraejemplo en caso contrario.
4. (2.5 pt.) Resolver el problema de la mochila 0 – 1, con los siguientes valores: Tamaño de la mochila  $M = 61$ ; número de objetos  $n = 5$ ; matriz de pesos  $W = (1, 11, 21, 23, 33)$ ; matriz de ganancias  $P = (11, 21, 31, 33, 43)$ .

[0.5 pt.] ¿Qué funciones de acotación utilizarías para reducir el espacio de búsqueda?

Representar el árbol de estados que se obtendría al utilizar las técnicas :

[1.00 pt.] Backtracking.

[1.00 pt.] Branch and Bound.

**IMPORTANTE:** En ambos casos, numerar los nodos según el orden en que son expandidos y comentar los criterios que se siguen para la expansión o poda de los nodos.
5. (2.5 pt.) Resolver el problema de encontrar los caminos mínimos entre cualquier par de puntos utilizando la técnica de la Programación Dinámica. Indicar la eficiencia del algoritmo propuesto e ilustrar el funcionamiento del mismo considerando un ejemplo con 5 vértices.