

Examen de Algorítmica. Septiembre de 2013. Tiempo disponible:  
Una hora y media.

1. Calcular  $t(n)$  y el orden de complejidad para

$$t(n) = 2t(n - 1) + 3^n(n + 1)$$

2. En el problema de la mochila 0/1 disponemos de tres mochilas, con capacidades  $M1$ ,  $M2$  y  $M3$ . El objetivo es maximizar la suma de beneficios de los objetos transportados en las tres mochilas, respetando las capacidades de cada una. Resolver el problema mediante programación dinámica, definiendo la ecuación de recurrencia, las tablas usadas y el algoritmo para rellenarlas.
3. Escribir un algoritmo de Backtracking para el problema de encontrar un subconjunto del conjunto  $T = \{t_1, t_2, \dots, t_n\}$ , que sume exactamente una cantidad  $P$ . Es necesario describir la representación de la solución, el esquema general del algoritmo, y las funciones de dicho esquema general.
4. En el problema de asignación, existen  $n$  trabajos y  $n$  personas. Cada persona  $i$  puede realizar un trabajo  $j$  con más o menos rendimiento:  $B[i, j]$ . El objetivo es asignar una tarea a cada trabajador (asignación uno-a-uno), de manera que se maximice la suma de rendimientos.

Para el problema de asignación de  $n$  tareas a  $n$  personas resuelto usando Ramificación y Poda, describir en primer lugar la representación de la solución, y en segundo lugar el método de obtención de estimaciones precisas para las funciones dadas por la Cota Inferior, la Cota Superior, y el Beneficio Estimado que emplearemos en las estrategias de poda y de ramificación. Es necesario mostrar el código de las funciones.