

Entregar cada ejercicio en una hoja distinta
No hace falta entregar esta hoja con el examen

1. (2 puntos) Dada la función:

```
operación Pares (a: array [1,...,n] de entero; izq, der: entero) : entero  
var m: entero  
m:= (der+izq)/2  
si m ≠ izq entonces  
    si par(a[m]) entonces  
        devolver Pares(a, izq, m) + 1  
    finsi  
    devolver 1  
sino  
    devolver 0  
finsi
```

Obtener razonando y justificando las respuestas:

- a) Las entradas en los casos más favorable y más desfavorable.
 - b) Unos buenos valores para Ω y O del tiempo de ejecución. ¿Qué ocurre con Θ ?
 - c) Ω y Θ del tiempo promedio.
 - d) Una fórmula de la ocupación máxima de memoria. Se supone que todos los parámetros se pasan por valor.
2. (1 punto) Elegir y resolver **una** de las siguientes cuestiones:
- a) Obtener la fórmula del tiempo de ejecución, en función del tamaño del problema y del caso base, de un algoritmo recursivo cuyo tiempo tiene la siguiente forma:
$$t(n) = 3 t(n-6) + 2, \text{ si } n > b; \quad t(n) = n^2, \text{ si } n \leq b.$$

Ojo, se pide la fórmula del tiempo, no sólo la notación asintótica.
 - b) Un alumno tiene que aprobar un conjunto de asignaturas, con sus exámenes, prácticas, entrevistas y demás. En total tiene que hacer **n** tareas. Para cada una de ellas estima que le llevará cierto tiempo, **t_i**. Como está un poco apurado de tiempo, se las piensa repartir entre las convocatorias de junio, septiembre y diciembre. En cada convocatoria puede sacar **M** unidades de tiempo. Se supone que todas las tareas se deben hacer antes o después, que las tareas no se pueden fraccionar (cada tarea va a una sola convocatoria), y que la suma de tiempos asignados a cada convocatoria no puede superar **M**.
El objetivo es conseguir un reparto haciendo las tareas cuanto antes, en vez de dejarlas para el final. Es decir, minimizar el tiempo dedicado en la convocatoria de diciembre. En caso de empate en diciembre, minimizar el tiempo en la de septiembre.
Resolver el problema mediante algoritmos voraces, aunque no se garantice siempre la solución óptima. Proponer y contrastar por lo menos dos criterios de selección diferentes. Aplicar el algoritmo al siguiente ejemplo: **n** = 5, **M** = 10, **t** = {5, 3, 1, 3, 4}.
3. (2,5 puntos) Resolver el problema del ejercicio 2b) mediante programación dinámica, definiendo la ecuación recurrente, con sus casos base, las tablas usadas y el algoritmo para rellenarlas. No es necesario mostrar la ejecución del ejemplo.
4. (2,5 puntos) Resolver el problema del ejercicio 2b) de forma óptima mediante un algoritmo de backtracking o mediante ramificación y poda. En cualquiera de los casos, utilizar los esquemas algorítmicos vistos en clase y explicar cómo es la representación de la solución, la forma del árbol y las funciones genéricas del esquema correspondiente. Se pueden suponer los esquemas vistos en clase, por lo que no hace falta repetirlos.

5. (2 puntos) Cierta problema de maximización se resuelve utilizando ramificación y poda, aplicando una estrategia LC-LIFO. La solución se representa mediante una tupla de tamaño 4, $\mathbf{s} = (s_1, s_2, s_3, s_4)$, donde cada s_i puede tomar valor 0 ó 1. Son soluciones todas las tuplas de 4 elementos en las que el número de unos sea menor o igual que 3. Mostrar el árbol generado en la ejecución del algoritmo, indicando también la evolución de la lista de nodos vivos, la variable de poda C, así como el orden de recorrido de los nodos, los nodos podados y la solución final. Suponer que para cada tupla se calculan las cotas y estimaciones indicadas abajo. ¿Cambiaría el resultado, en este ejemplo, si se utilizara la estrategia LC-FIFO? Justifica la respuesta de forma muy breve.

Tupla	CI	BE	CS
Raíz	2	6	13
0	2	7	10
1	3	8	12
00	8	8	8
01	4	7	10
10	5	7	10
11	6	6	12
000	4	6	7
001	8	8	8
010	6	7	8
011	3	4	7

Tupla	CI	BE	CS
100	6	7	10
101	4	8	8
110	4	5	9
111	5	5	11
0000	6	6	6
0001	5	5	5
0010	2	2	2
0011	8	8	8
0100	1	1	1
0101	7	7	7
0110	4	4	4

Tupla	CI	BE	CS
0111	3	3	3
1000	3	3	3
1001	10	10	10
1010	5	5	5
1011	7	7	7
1100	8	8	8
1101	9	9	9
1110	11	11	11
1111	9	9	9

Nota: La pregunta 5 no deben hacerla los alumnos que tengan convalidada la Práctica 4. Los alumnos de "Laboratorio de programación" tienen convalidada esta práctica con la nota que obtuvieran en dicha asignatura.