APELLIDOS, NOMBRE_____ **NOTA**_____

Nº mesa_____ Tiempo máximo **120** minutos

Ejercicio_1. (2 puntos)

La ecuación de recurrencia de abajo surge del análisis de cierto algoritmo que usa divide y vencerás.

$$f(n) = \begin{cases} 4n & \text{Si } n \leq 3 \\ n^2 + 6 & \text{Si } 3 < n \leq 10 \\ 2\,f(n/2) + 3\,f(n/4) & \text{En otro caso} \end{cases}$$

1) (1 punto) Encontrar una fórmula no recursiva para el valor de **f(n)**, calculando también el valor de las constantes que puedan aparecer. Expresar la complejidad de **f(n)** usando las notaciones asintóticas O, Θ y Ω.
2) (0.5 puntos) Escribir un procedimiento en seudocódigo (incluida la cabecera) cuyo análisis de complejidad pueda dar como resultado la anterior ecuación. Indicar cuál es la unidad medida por **f(n)** (nº de instrucciones, segundos, ...)
3) (0.5 puntos) Suponer que en el caso de abajo cambiamos n/4 por n/3, teniendo:
   f´(n) = 2f´(n/2) + 3f´(n/3). Demostrar por inducción que $O(f(n)) \subseteq O(f'(n))$.

Ejercicio_2. (3 puntos)

En el problema de la mochila para el caso en que no se permita partir los objetos (es decir, un objeto se coge entero o no se coge nada) tenemos:

- **n** objetos, cada uno con un peso ($\mathbf{p_i}$) y un valor o beneficio ($\mathbf{b_i}$)
- Una mochila en la que podemos meter objetos, con una capacidad de peso máximo **M**.
- ☐ **Objetivo:** llenar la mochila con esos objetos, maximizando la suma de los beneficios (valores) transportados, y respetando la limitación de capacidad máxima **M**.
- ☐ Se supondrá que los objetos **NO** se pueden partir en trozos.

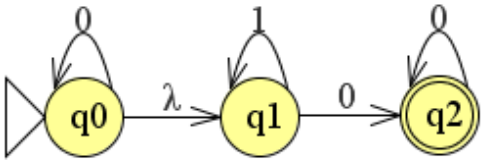
1) (1 punto). Diseñar un algoritmo **voraz** para resolver el problema, aunque no se garantice la solución óptima. Es necesario marcar en el código propuesto a que corresponde cada parte en el esquema general de un algoritmo voraz (criterio, candidatos, función……). Si hay más de un criterio posible elegir uno razonadamente y discutir los otros. Comprobar si el algoritmo garantiza la solución óptima en este caso (la demostración se puede hacer con un contraejemplo).
   - Aplicar el algoritmo al caso: **n**= 3, **M**= 6, **p**= (2, 3, 4), **b**= (1, 2, 5)
2) (1 punto). Resolver el problema mediante **programación dinámica**. Definir la ecuación recurrente, los casos base, las tablas y el algoritmo para rellenarlas y especificar cómo se **recompone la solución** final a partir de los valores de las tablas.
   - Aplicar el algoritmo al caso: **n**= 3, **M**= 6, **p**= (2, 3, 4), **b**= (1, 2, 5)
3) (1 punto). Resolver el problema por backtracking usando el esquema iterativo. Indicar cómo es la representación de la solución, la forma del árbol y programar las funciones genéricas del esquema correspondiente.
   - Aplicar el algoritmo al caso: **n**= 3, **M**= 6, **p**= (2, 3, 4), **b**= (1, 2, 5)

APELLIDOS, NOMBRE_____ **NOTA**_____

Nº mesa_____ Tiempo máximo **120** minutos

Ejercicio_3. (2 puntos)

Dado el AFND



1) (0.25 puntos). Especifica el Lenguaje aceptado por el automata y todas las cadenas de longitud igual a tres aceptadas por el mismo.
2) (1 punto). El A.F.D. mínimo equivalente.
3) (0.5 puntos). La expresión regular del lenguaje reconocido por el autómata del apartado anterior.
4) (0.25 puntos). La gramática de tipo 3 para el lenguaje.

Ejercicio_4. (3 puntos)

Dado el lenguaje libre de contexto:

$$L = \{x^n y^{2n} z \,/\, n \geq 0\}$$

1) (0.5 pto.) Construir una gramática LL(1) que lo genere.
2) (0.25 pto.) Obtener un *Autómata con Pila* asociado al lenguaje que acepte el mismo lenguaje por pila vacía.
3) (0.5 pto.) Analizar por el autómata anterior, teniendo en cuenta el principio de preanálisis (lectura de un símbolo de la entrada con anticipación) las entradas **"xyyz"** y **"xyz"**.
4) (0.5 pto.) Con la gramática LL(1), construir la tabla de análisis LL(1) y especificar el pseudocódigo de análisis sintáctico tabular.
5) (0.5 pto.) Construir las trazas correspondientes al reconocimiento de las frases: **"xyyz"** y **"xyz"** según el pseudocódigo especificado en el punto 4 anterior.
6) (0.75 pto.) Especificar el pseudocódigo de análisis sintáctico LL dirigido por la sintaxis. Analizar si son correctas sintácticamente las entradas: **"xyyz"** y **"xyz"**