

**Ejercicio\_1.** (2 puntos)

Se tiene un vector de enteros no repetidos y ordenados de menor a mayor. Diseñar un algoritmo que compruebe en tiempo logarítmico si existe algún elemento del vector que coincida con su índice. Se pide:

- (0.5 puntos). Elección del esquema (voraz, vuelta atrás, divide y vencerás). Identificación del problema con el esquema.
- (1 punto). Estructura de datos. Algoritmo completo (con pseudocódigo). Aplicar a  $v = \{-3, -2, 0, 2, 3, 5, 7, 9, 12\}$
- (0.5 puntos). Estudio del coste.

**Ejercicio\_2.** (3 puntos)

➤ Resolver 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).

□ Problema de la mochila: Una mochila en la que podemos meter objetos, con una capacidad de peso máximo  $M$ ; tenemos  $n$  objetos, cada uno con un peso ( $p_i$ ) y un valor o beneficio ( $b_i$ ). El **Objetivo** es 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.

➤ Se pide:

- (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:  $n=3$ ,  $M=6$ ,  $p=(2, 3, 4)$ ,  $b=(1, 2, 5)$
- (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:  $n=3$ ,  $M=6$ ,  $p=(2, 3, 4)$ ,  $b=(1, 2, 5)$
- (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:  $n=3$ ,  $M=6$ ,  $p=(2, 3, 4)$ ,  $b=(1, 2, 5)$

**Ejercicio\_3.** (2 puntos)

Una caja fuerte dispone de una única cerradura. Para abrirla es necesario hacer girar en ella tres llaves diferentes (denominadas 1, 2 y 3), en un orden predeterminado, que se describe a continuación:

- Llave 1, seguida de llave 2, seguida de llave 3, o bien
- Llave 2, seguida de llave 1, seguida de llave 3.

Si no se respeta este orden, la caja se bloquea, y es imposible su apertura; por ejemplo, si se hace girar la llave 1, se retira la misma, se introduce de nuevo y se hace girar. Una vez abierta la caja fuerte, la introducción de las llaves en su cerradura, en cualquier orden, no afecta al mecanismo de cierre (permanece abierta).

Considérese que las denominaciones de las llaves son símbolos de un alfabeto, sobre el que define el lenguaje  $L$  cuyas palabras son las secuencias permitidas de apertura de la caja fuerte. Por ejemplo, 12323 es una palabra del referido lenguaje.

Se pide:

- (1 punto). Diseño del autómata AFND que acepta el lenguaje  $L$ .
- (0.5 puntos). Gramática que genera las palabras de  $L$ .
- (0.5 puntos). Autómata AFD mínimo equivalente del apartado **a**.

**Ejercicio\_4.** (3 puntos)

Considérese la siguiente gramática:

$S \rightarrow S = A \mid A$

$A \rightarrow A + B \mid A - B \mid B$

$B \rightarrow (S) \mid a \mid b$

- (0,5 puntos). Eliminar la recursividad por la izquierda y comprobar si es LL(1).
- (0.5 puntos). Con la gramática equivalente LL(1), especificar un autómata con pila que acepte el mismo lenguaje por pila vacía. Analizar por el autómata anterior, teniendo en cuenta el principio de preanálisis (lectura de un símbolo de la entrada con anticipación) la entrada " $a = b$ ".
- (1 punto) Con la gramática equivalente LL(1), construir la tabla de análisis LL(1) y especificar el pseudocódigo de análisis sintáctico tabular. Construir la traza correspondiente al reconocimiento de la frase: " $a = b$ " según el pseudocódigo especificado.
- (1 punto) Especificar el pseudocódigo de análisis sintáctico dirigido por la sintaxis para la gramática obtenida LL(1) y realizar el análisis para la entrada " $a = b$ ".