

No hay que entregar esta hoja con el examen.

1. (2 puntos) Partimos de una especificación formal, usando el método axiomático, del TAD **Conjunto[T]**, suponiendo los siguientes tipos y operaciones predefinidos:

- **Conjunto**: vacío, esVacio, inserta, suprime, esMiembro, cardinal, unión, intersección, diferencia.
- **Natural**: cero, esCero, sucesor, suma, esIgual, resta.

Queremos añadir las siguientes operaciones:

- **invierte**: dado un conjunto y un elemento, si el elemento pertenece al conjunto entonces lo suprime; si no pertenece, lo inserta. Especificar esta operación sin usar la operación esMiembro.
- **mayoría**: dados tres conjuntos, devuelve un conjunto con todos los elementos que están al menos en dos o más conjuntos de los tres de entrada.
- **minoría**: dados tres conjuntos, devuelve un conjunto con todos los elementos que están sólo en uno de los tres conjuntos de entrada.
- **potencia**: dado un conjunto **c**, devuelve un conjunto con todos los subconjuntos posibles de **c** (es decir, el resultado sería un conjunto de conjuntos).

Escribir la sintaxis y la semántica de las operaciones añadidas. Se pueden añadir otras operaciones, que deberán ser especificadas también.

Nota: La pregunta 1 no deben hacerla los alumnos que tengan aprobada la práctica de especificaciones formales.

2. (2,5 puntos) Suponer que tenemos implementada una estructura de árboles trie. Queremos añadir información a cada nodo del trie, que indique el número de palabras descendientes de ese prefijo. Si **n** es un nodo, el valor se almacenará en el atributo **n.numDesc**.

Escribir una operación que, dada la raíz de un trie, calcule el número de palabras descendientes para todos los nodos del árbol (es decir, que calcule y ponga los valores **n.numDesc** de todos los nodos). Programarla utilizando exclusivamente las operaciones genéricas sobre nodos trie: **Consulta (n: trie, c: carácter): trie**, **Inserta (n: trie, c: carácter, m: trie)**, **NuevoTrie: trie** y un iterador del tipo **para cada carácter c hijo del nodo n hacer**.

3. (2,5 puntos) Sobre un árbol **B** de orden **p = 5** se insertan los siguientes elementos: 72, 21, 10, 84, 23, 81, 28, 53, 62. Después se eliminan los valores 81 y 21. Mostrar la evolución del árbol y el resultado final.

4. (3 puntos) Escribir un algoritmo de orden $O(n)$ para determinar si un grafo de **n** vértices tiene algún ciclo. En caso afirmativo, el algoritmo debe mostrar el ciclo encontrado. Justificar el hecho de que el orden del algoritmo es un $O(n)$.

No hay que entregar esta hoja con el examen.

1. (3 puntos) El conteo de instrucciones de cierto algoritmo recursivo da como resultado la siguiente ecuación:

$$t(n) = \begin{cases} 8 & \text{Si } n < 10 \\ 2 \cdot t(n/3) + 3 \cdot t(n/9) + 5n + 3 & \text{En otro caso} \end{cases}$$

Escribe un algoritmo cuyo tiempo sea dado por la ecuación $t(n)$. Calcular el tiempo de ejecución, el orden de complejidad y el uso de memoria del algoritmo.

2. (2 puntos) Curro se ha ido de vacaciones al Caribe. Desde allí quiere mandar postales a sus allegados. En una oficina de correos de Cancún ha comprado un lote de sellos de n valores diferentes, tres sellos de cada valor. En correos le han informado de que la tarifa para España es T , y le han explicado que debe pegar exactamente 5 sellos en la postal. Por lo tanto, un franqueo sólo es admisible si se alcanza la tarifa (es decir, valor T o mayor) y se ponen exactamente cinco sellos.

Diseñar un algoritmo voraz que encuentre una buena forma de resolver el problema, es decir un franqueo admisible con valor T o lo más próximo posible a T . Suponer que los costes de los n tipos de sellos vienen dados en el array $c = (c_1, c_2, \dots, c_n)$.

3. (2,5 puntos) Resolver el problema del ejercicio 2 de forma óptima por programación dinámica. Queremos saber si existe una forma de obtener valor exacto T usando 5 sellos. Dar la fórmula recursiva del problema, con sus casos base, e indicar las tablas que usa el algoritmo. Ojo: tener en cuenta que aparece una restricción por la cantidad a alcanzar y otra por el número de sellos a utilizar.
4. (2,5 puntos) Resolver el problema del ejercicio 2 de forma óptima por backtracking. Se deberán utilizar los esquemas vistos en clase, que se pueden dar por supuestos. Definir la forma de representar la solución y las demás funciones genéricas del esquema.

Nota: Los alumnos que tengan aprobada la práctica 4 tienen convalidado el ejercicio 2.