Examen. 24 de enero de 2017

Todas las preguntas tienen la misma ponderación (25%).

1. Dado el TAD *Pila* de elementos de tipo *T* con las operaciones *crear*, *esVacía*, *push*, *pop*, *tope*, escribir su especificación formal algebraica añadiendo las siguientes operaciones:

   a) *fondo*: devuelve el elemento del fondo de la pila;
   b) *quitarFondo*: devuelve la pila sin el elemento del fondo;
   c) *capicúa*: dada una pila, determina si es capicúa o no.

   Nota: se pueden añadir otras operaciones si se consideran necesarias.

2. Disponemos de una imagen de 1.600x1.250 píxeles. Dentro de ella, se eligen ciertos píxeles al azar y se almacena cierta información para esos píxeles concretos en una tabla de dispersión. Disponemos de una tabla de 4.000 cubetas. Se pide:

   a) Si usamos dispersión cerrada, discutir sobre cuál sería el número de píxeles que se pueden almacenar de forma que el comportamiento de la tabla fuera razonablemente eficiente. ¿Y si usamos dispersión abierta? Razonar las respuestas incluyendo una representación gráfica de la eficiencia.
   b) Diseñar una buena función de dispersión para esta aplicación. Justificar la decisión, indicando cómo son las claves en este problema.
   c) Suponer que tenemos almacenados 65.536 valores en la tabla. Estimar el número promedio de operaciones en la consulta, con dispersión abierta y con dispersión cerrada. Hacer las consideraciones oportunas sobre la estrategia de reestructuración.

3. En un conjunto de palabras representado con estructuras de árboles, queremos añadir una operación que encuentre de forma eficiente todas las palabras que empiezan por la letra D, luego una vocal cualquiera, y luego pueden aparecer otras letras cualesquiera. Elegir la estructura que se prefiera, entre árboles B, AVL o trie, y escribir la operación en pseudocódigo. Si se hace con árboles trie no hay que suponer una representación concreta de los nodos sino usar la consulta sobre el tipo nodo. En cualquier caso, razonar qué ocurriría si lo que buscamos son las palabras que acaben en D + vocal.

4. Una salmodia es un canto iterado, que consta de una serie de versos que se repiten indefinidamente. Queremos crear una salmodia. Para ello tenemos un conjunto de *n* versos. Tenemos una matriz de booleanos *M*, en la que para cada par de versos, (*i*, *j*), sabemos si después del verso *i* puede ir o no el *j*. Por otro lado, sabemos la longitud de cada verso, *L[i]*. Escribir un algoritmo que encuentre de forma eficiente la salmodia más corta posible empezando en el verso 1, es decir, una sucesión de versos, teniendo en cuenta que después del último viene otra vez el primero.