

# Estructuras de Datos y Algoritmos

Examen 12/12/2011

---

Nombre y Apellido

C.I.

Turno

Por favor siga las siguientes indicaciones:

- Escriba con lápiz.
- Escriba su nombre y c.i. en todas las hojas que entregue.
- Numere las hojas e indique el total de hojas en la primera de ellas.
- Escriba las hojas de un solo lado.
- Comience cada ejercicio en una hoja nueva.
- El total máximo de puntos del examen es de **100**.
- Se necesitan 60 puntos para aprobar.
- El parcial contiene un total de: **3 carillas**.
- La prueba es individual y sin material.
- Solo se contestan dudas acerca de letra de los ejercicios.

## Ejercicio 1

Considere la siguiente declaración, en C, del tipo de los nodos de listas dinámicas de enteros:

```
struct nodoLista {  
    int info;  
    nodoLista *sig; };
```

Implemente en C una función iterativa **F** que dadas dos listas ordenadas (en forma ascendente) de enteros, **C1** y **C2**, y sin elementos repetidos que representan a dos conjuntos, retorne una nueva lista ordenada y sin elementos repetidos que represente al conjunto  $(C1-C2) \cup (C2-C1)$ .

A tener en cuenta:

- La nueva lista no deberá compartir registros de memoria con las listas parámetros.
- Se requiere que esta función recorra a lo sumo una vez cada lista parámetro.
- No se permite usar TADs auxiliares.
- Si usa funciones o procedimientos auxiliares deberá implementarlos, pero recuerde que cada una de las listas pueden recorrerse a lo sumo una vez (se usen o no funciones o procedimientos auxiliares).

La firma de la función es la siguiente: `nodoLista * F (nodoLista * C1, nodoLista * C2) ;`

## Ejercicio 2

**Parte a)** Implemente un procedimiento recursivo *borrarMayores* que dados un árbol binario de búsqueda de enteros **A** de tipo **ABB** (sin elementos repetidos) y un entero **x**, elimine de **A** todos los elementos mayores a **x**. El árbol resultado debe ser un árbol binario de búsqueda.

**NOTAS:** Se sugiere usar un procedimiento auxiliar, que deberá ser implementado, que elimine todos los nodos de un árbol, dejando a éste vacío. No se permite definir procedimientos ni funciones auxiliares, además del procedimiento sugerido. Tampoco se pueden definir estructuras de datos auxiliares. No se permite recorrer el árbol más de una vez. Deberá liberarse la memoria de todas las celdas cuyos elementos sean eliminados.

**Parte b)** ¿Cuál es el orden (**O**) de tiempo de ejecución en el peor caso del procedimiento *borrarMayores*?. Explique breve e informalmente.

### Ejercicio 3

Se quiere especificar e implementar un TAD *ColaDoble* que permita insertar y eliminar elementos (de a uno por vez) al comienzo y al final de una estructura lineal. Se pide:

- a) Especificar, incluyendo pre y postcondiciones, el TAD *ColaDoble* de elementos de un tipo genérico adecuado para modelar una estructura lineal no acotada, con operaciones que permitan:
  - Crear la estructura vacía,
  - Agregar un elemento al comienzo de la estructura lineal,
  - Agregar un elemento al final de la estructura lineal,
  - Testear por la estructura vacía,
  - Eliminar y retornar el elemento al comienzo de la estructura lineal (si la estructura es no vacía),
  - Eliminar y retornar el elemento al final de la estructura lineal (si la estructura es no vacía),
  
- b) Implementar el TAD anterior, sin usar TADs auxiliares, de tal manera que todas las operaciones (con excepción de la que crea la estructura vacía) sean  $O(1)$  peor caso. Desarrolle el código de las siguientes 3 operaciones: la operación que crea la estructura vacía; la que elimina y retorna un elemento al final; y, la que agrega un elemento al comienzo. Omita el código del resto de las operaciones del TAD.