

Estructuras de Datos y Algoritmos

Examen 20/12/2012

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(35 Puntos)

Considere la siguiente declaración, en C/C++, del tipo *ListaEnt* de listas encadenadas de enteros:

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

- a) Implemente una función ***intersección*** que dadas dos listas de ListaEnt ordenadas, devuelva una tercera lista con los elementos que están en la intersección de las listas pasadas como parámetro. La lista resultado no debe compartir memoria con las listas pasadas como parámetro.

Nota: No se pueden utilizar funciones auxiliares ni recorrer la lista parámetro mas de una vez.

- b) ¿Cuál es el orden de tiempo de ejecución en el peor caso de las operación de la parte a). Justifique.
- c) Si no estuvieran ordenadas las listas parámetro ¿cambia el orden de tiempo de ejecución?

Ejercicio 2(35 Puntos)

Considere las siguientes declaraciones, en C/C++, del tipo de los árboles binarios de búsqueda de enteros, de tipo ABB:

```
struct nodoABB {  
    int dato;  
    int altura;  
    ABB izq;  
    ABB der;  
};  
typedef nodoABB *ABB;
```

Se pide:

- a) Implementar una función que dado un ABB de enteros que guarda en cada nodo, además del dato, el valor de su altura, y dado un entero, inserte a dicho elemento en el ABB (asuma que el elemento no pertenece al ABB) manteniendo la información de la altura de cada nodo. Insertar el elemento en el ABB dejando en cada nodo en el campo altura el valor correspondiente, asumiendo que el árbol parámetro guarda en el campo altura de cada nodo el valor correcto.

- b) Implemente una función que retorne TRUE si y sólo si un ABB (del tipo definido anteriormente) es un AVL. Esto es, que para cada nodo del árbol la altura de sus subárboles izquierdo y derecho difiere a lo sumo en 1. El árbol vacío es un AVL.

Nota: No se pueden utilizar funciones auxiliares ni recorrer el ABB mas de una vez.

Ejercicio 3(35 Puntos)

Se desea implementar un Stack (Pila) no acotado que almacene valores enteros.

- a) Defina un TAD que permite que resuelva el problema. Indicando pre y post condiciones para cada operación.
- b) Implemente las operaciones de inserción y extracción.
- c) Dado que trabajamos en C/C++ necesitamos liberar la memoria, (en otros lenguajes como Java el programador no se encarga de eso). Implemente **sin acceder a la representación**, una función que permita destruir un Stack sin dejar memoria colgada (sugerencia: utilice las funciones que definió en la parte a).