Programación y Estructuras de Datos (PED)

Examen sobre prácticas Julio 2020

Condiciones de entrega

- El examen se entrega a través del servidor de prácticas del DLSI http://pracdlsi.dlsi.ua.es**. Tras cada entrega, el servidor enviará al alumno un INFORME DE COMPILACIÓN**, para que el alumno compruebe que lo que ha entregado cumple las especificaciones pedidas y que se ha podido generar el ejecutable correctamente. **Este informe también se podrá consultar desde la página web de entrega de prácticas del DLSI** (http://pracdlsi.dlsi.ua.es *e introducir el nombre de usuario y password*). **SE ACONSEJA** que se haga una entrega previa con al menos el prototipo de la función pedida, para comprobar que no hayan errores de formato. Es posible que el servidor tarde más tiempo en devolver los informes en la última media hora del examen: **LO CUAL NO SERÁ ARGUMENTO** para justificar errores
- **En caso que la práctica esté correctamente entregada, compilada y ejecutada,** en este informe debe salir lo siguiente:

```
================================================
 DIFERENCIA CON FICHERO DE SALIDA DE REFERENCIA
================================================

----------------------------------------
```

- En el servidor, el examen se comprobará con la siguiente función main:

```
TAVLCom A; TListaCom L; TVectorCom  V;
L = A.Caminos_AVL(V);
cout << L << endl;
```

- Esperando la salida:

```
{}
```

- Se tiene que entregar un fichero comprimido **tgz** (**tar cvzf fichero.tgz ***) que contenga todos los ficheros de los cuadernillos (con la estructura de directorios especificada en el enunciado de la práctica: **dentro del .tgz solo deben aparecer el fichero nombres.txt y los directorios lib, include y src**), junto con los métodos pedidos en el examen. El examen debe compilar con todos los ficheros entregados.
- El fichero **nombres.txt** tiene que contener el nombre del único autor del examen.
- El nombre de la función implementada por el alumno debe coincidir EXACTAMENTE con el prototipo propuesto en el enunciado.
- El alumno tiene que implementar su propio fichero de prueba (**tad.cpp**) para comprobar el código implementado (**este fichero no es necesario entregarlo**). Tampoco es necesario que se entregue el fichero makefile.
- El alumno **puede añadir a la parte privada las variables y métodos** que considere necesarios para la implementación.
- **SI SE ENTREGA ALGO QUE NO COMPILA SUPONDRÁ UN CERO EN EL EXAMEN. Solo se evaluará la salida del programa. Se compilará con la versión del compilador instalada en los laboratorios de la EPS.**
    - **ARCHIVOS A ENTREGAR (incluyendo en el código la función Caminos_AVL) :**

```
include: tcomplejo.h, tvectorcom.h, tlistacom.h, tavlcom.h, tabbcom.h
lib: tcomplejo.cpp, tvectorcom.cpp, tlistacom.cpp, tavlcom.cpp, tabbcom.cpp
```

TListaCom **Caminos_AVL** (TVectorCom &)

El método **Caminos_AVL** invocado por un árbol AVL **TAVLCom** , devuelve una lista de TComplejo (**TListaCom**) y recibe un vector de TComplejo (**TVectorCom**).

Hay que:

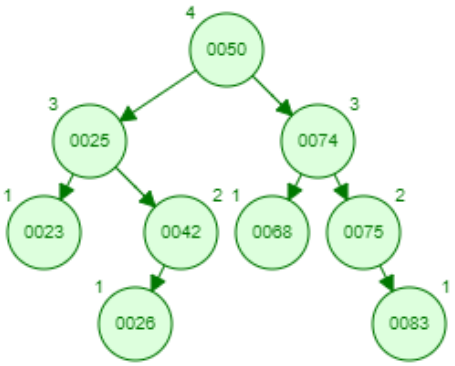
- Recorrer el TVectorCom de izquierda a derecha.
- Buscar cada elemento TComplejo del vector, en el árbol TAVLCom que invoca al método.
- Si el TComplejo chequeado existe en el árbol TAVLCom, operar así, según sea su PARTE REAL:
    - Si su PARTE REAL es PAR → insertar al final de la lista TListaCom los elementos TComplejo de cada nodo del CAMINO que va desde el TComplejo chequeado (inclusive), hasta el nodo HOJA correspondiente, escogiendo el hijo DERECHO cuando haya 2 a elegir; si no hay 2 a elegir, se elige el único hijo posible.
    - Si su PARTE REAL es IMPAR → insertar al final de la lista TListaCom los elementos TComplejo de cada nodo del CAMINO que va desde el TComplejo chequeado (inclusive), hasta el nodo HOJA correspondiente, escogiendo el hijo IZQUIERDO cuando haya 2 a elegir; si no hay 2 a elegir, se elige el único hijo posible.

- Si el TComplejo chequeado no existe en el árbol TAVLCom, insertar al final de la lista TListaCom un elemento TComplejo vacío.

NOTAS:

- Si el vector TVectorCom está vacío, se devolverá una lista TListaCom vacía (ningún TComplejo a chequear).
- Si el árbol TAVLCom está vacío y el vector no, se devolverá una lista TListaCom con tantos elementos vacíos como elementos tenga el vector de entrada.

Ejemplos (se muestra sólo la PARTE REAL de cada TComplejo, por simplificación):



Ejemplo 1:

VECTOR ENTRADA: [50, 23, 75, 42, 25, 68, 74, 83, 26]

LISTA SALIDA: {50, 74, 75, 83, 23, 75, 83, 42, 26, 25, 23, 68, 74, 75, 83, 83, 26}

Ejemplo 2:

VECTOR ENTRADA: [50, 25, 74, 70]

LISTA SALIDA: {50, 74, 75, 83, 25, 23, 74, 75, 83, vacío}

Ejemplo 3:

VECTOR ENTRADA: [51, 42, 70]

LISTA SALIDA: {vacío, 42, 26, vacío}