

Estructuras de Datos – Cursado 2019

EJERCICIOS SIMULACRO PRIMER PARCIAL

El parcial incluirá 3 incisos, uno de cada tipo

EJERCICIO TIPO 1: Implementar una operación de un TDA lineal

- a) Agregar al TDA Lista la operación *obtenerMultiplos(int num)* que recibe un número y devuelve una lista nueva que contiene todos los elementos de las posiciones múltiplos de num, en el mismo orden encontrado, **haciendo un único recorrido de las estructuras** original y copia; y sin usar otras operaciones del TDA. **Ejemplo:** si se invoca con la lista <A,B,C,D,E,F,G,H,I,J> y num=3, el método debe devolver la lista <C,F,I>
- b) Agregar al TDA Lista la operación *eliminarApariciones(TipoElemento x)* que elimine todas las apariciones de elementos iguales a x, **haciendo un único recorrido de la estructura** y sin usar otras operaciones del TDA.
- **Consideraciones ejercicio 1 a y b:**
 - Realizar la definición de tipos de todas las clases involucradas
 - En todas las operaciones recorrer lo menos posible las estructuras

EJERCICIO TIPO 2: Usar TDA lineales

- c) En una clase TestCadenas, que utilice los TDA Lista, Pila y Cola vistos en la materia, para guardar elementos de tipo CHAR, implementar el método **generar (Cola c1)** que recibe por parámetro una estructura cola **c1** que tiene el siguiente formato: $a_1\#a_2\#a_3\#\dots\#a_n$, donde cada a_i en una sucesión de letras mayúsculas y a partir de **c1** debe generar como salida otra Cola de la forma: $a_1a_1'a_1\#a_2a_2'a_2\#\dots\#a_na_n'a_n$ donde cada a_i' es la secuencia de letras mayúsculas a_i pero invertida. Ejemplo.: Si **c1** es : AB#C#DEF , entonces la operación **generar** devolverá una Cola con el siguiente formato: ABBAAB#CCC#DEFFEDDEF.
- d) En la clase Matematica, que utiliza a los TDA Lista, Pila y Cola vistos en la materia para guardar elementos tipo CHAR que representan una expresión matemática, desarrollar el método **verificarBalanceo (Cola q)** que recibe por parámetro una cola con una expresión matemática y verifique que los paréntesis, corchetes y llaves estén correctamente balanceados. Debe usar como estructura auxiliar alguno de los TDA lineales vistos, el que considere más apropiado.
Ejemplos: Si q es $\leftarrow \{ 5 + [8 * 9 - (4 / 2) + 7] - 1 \} \leftarrow$ el método debe devolver **TRUE**
Si q es $\leftarrow \{ 5 + 8 * 9 - (4 / 2) + 7] - 1 \} \leftarrow$ el método debe devolver **FALSE**
- **Consideraciones ejercicio 2 c y d:**
 - Dibujar la cajas UML del TDA Lista, Pila y Cola indicando nombre, parámetros y tipo de salida de su interfaz (métodos públicos) y sólo las operaciones del TDA sin indicar la implementación.

EJERCICIO TIPO 3: Implementar una operación del TDA Arbol Binario

- e) Implementar la operación **boolean verificarPatron(Lista patron)**, que recibe por parámetro una lista patron y determine si coincide exactamente con al menos un camino del árbol que comience en la raíz y termine en una hoja. *El método debe ser eficiente, es decir, recorrer el árbol lo estrictamente necesario.*
- f) Implementar la operación **frontera()** que devuelve una lista con la secuencia formada por los elementos almacenados en las hojas del árbol binario, tomadas de izquierda a derecha.
- g) Implementar la operación **clonarInvertido()** que devuelve un nuevo árbol, que es una copia del árbol original (this) pero donde los hijos están cambiados de lugar. Atención: el método devuelve un nuevo árbol, sin modificar el árbol original.
Consideraciones:
 - El árbol debe estar implementado con nodos enlazados.
 - Escribir la definición de tipos de la clase ArbolBin y NodoArbol

PARA CADA MÉTODO IMPLEMENTADO:
(1) DECIR CUAL ES EL ORDEN DE EJECUCIÓN Y JUSTIFICAR
(2) REALIZAR AL MENOS UNA TRAZA CON LA ESTRUCTURA VACIA Y CON N ELEMENTOS