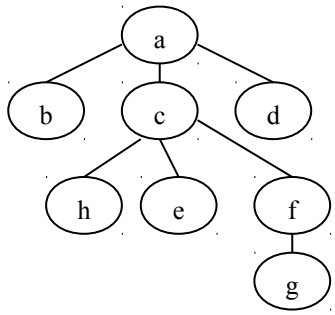
UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
ALGORITMOS Y ESTRUCTURAS DE DATOS

PRÁCTICA DE ÁRBOLES GENERALES Y BINARIOS

1. Usando las primitivas de árboles generales desarrolle una función que retorne la suma de todos los enteros comprendidos entre dos niveles *K* y *L* del árbol.
2. Digamos que un árbol es semi-completo cuando sus nodos no hojas, tienen el mismo grado. Empleando las primitivas de árbol general, elabora un algoritmo que permita saber si un árbol es semi-completo (el grado del árbol no se conoce).
3. Elabore los recorridos en inorden (también llamado simétrico) y preorden de árboles generales.
4. Dado un árbol genealógico representado por un árbol general, implemente la operación Cousin donde dado el nombre de una persona, imprimir a todos sus primos.
5. Usando las primitivas de árboles generales, implemente la operación PodarHojas, la cual consiste en eliminar todos aquellos nodos que son hojas en un árbol.
6. Elabore un algoritmo empleando las primitivas de árbol general, que retorne el siguiente valor:
   Valor(A) = Productoria(Left(A)) - Productoria(Right(A))
   Donde Productoria(Left(A)) es la productoria de todos nodos en el subárbol izquierdo de A, y Productoria(Right(A)) corresponde con a la productoria de los nodos en el subárbol derecho de A. Sólo debe recorrer una sola vez cada nodo, y el perfil de la función es el siguiente: **function** Valor (GenTree B) : Integer**.**
7. Considere la siguiente representación estática de árboles:

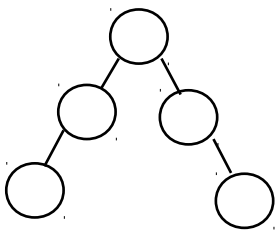


b	h	e	g	f	c	d	a	Info
0	0	0	0	1	3	0	3	Grado
1	2	3	4	5	6	7	8	Índice

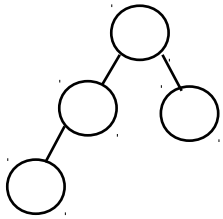
   Se puede ver que la forma de almacenamiento del árbol consiste en tener su recorrido en postorden definido en el arreglo (además de tener el grado de cada nodo). ¿Qué desventajas podría tener este enfoque de almacenamiento estático? Basada en esta representación defina las primitivas de GetChild, GetRBrother y GetParent. **Sugerencia**: Utilice una pila auxiliar.
8. Construya diferentes árboles binarios tal que sus nodos aparezcan exactamente en el mismo orden en los recorridos:
   a. Preorden y Simétrico b. Postorden y Simétrico c. Preorden y Postorden
9. Se tiene un lenguaje que no permite la recursión, y para simularla emplea una pila en la cual almacena en el caso de los árboles, el camino recorrido desde la raíz hasta el nodo visitado. Se quiere que Ud. realice el algoritmo que permita recorrer un árbol binario (Sin utilizar recursión) en Simétrico.
10. Usando las operaciones de BinaryTree, desarrolle las siguientes operaciones:
    a) **function** IsLeaf(BinaryTree T, BinaryTree X) : **Boolean**
       Tal que isLeaf(A,X) = verdad si y sólo sí Grado(GetRoot(X)) = 0 en el árbol T.
    b) **Function** IsParent(BinaryTree A, BinaryTree x, BinaryTree y) : **Boolean**
       Tal que IsParent(A,x,y) retorna **true** si y sólo si GetRoot(x) es ascendiente directo de GetRoot(y), y retorna **false** en caso contrario.

    ¿Qué modificaciones le haría a la estructura original para que IsParent sea O(1)?
11. Utilizando las primitivas de la clase BinaryTree elabore un algoritmo que verifique si un árbol binario es completo.
12. Un Árbol Binario S está incluido en un Árbol Binario T, si S es igual a T o si S es igual a un subárbol de T. Escriba una función que indique si un árbol S está incluido en un árbol T, en cuyo caso debe retornar la dirección del subárbol de T que es igual a S.
13. Escriba una función que verifique la simetría de un árbol binario cualquiera. Ejemplos:

Simétrico



No Simétrico



14. Se tiene la secuencia en preorden, postorden y simétrico de un árbol de n nodos almacenados en tres listas distintas. Realice un algoritmo que determine si el nodo i es ascendiente del nodo j (para cualquier par de nodos i, j). ¿Sería posible elaborar dicho algoritmo con sólo dos recorridos? Si su respuesta es afirmativa ¿Cuáles serían esos pares de recorridos, y elabore los

correspondientes algoritmos? Si su respuesta es negativa, justifique detalladamente el porqué. Elabore el (los) algoritmo (s) tanto para árboles binarios como para árboles generales

15. Dada una secuencia en notación prefija y almacenada como una SECUENCIA[ELEMENTO], se desea que Ud. implemente una operación utilizando las primitivas de Árboles binarios y del tipo Secuencia, que almacene dicha expresión en un árbol binario, tal que si se recorre ese árbol en Simétrico se obtenga la expresión en notación infija. Los elementos de la secuencia pueden ser operadores u operandos. Sobre dichos elementos se tiene definida la siguiente primitiva:

```
function IsOperator(ELEMENTO E) : Boolean
    if E ∈ { + , - , * , / } then
            return  (True)
    else
            return (False)
    end
end
```

GDAYED/2014