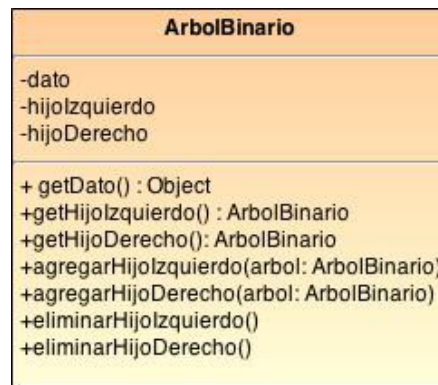


Práctica 2: Estructuras de Datos y Algoritmos II UPE 2015

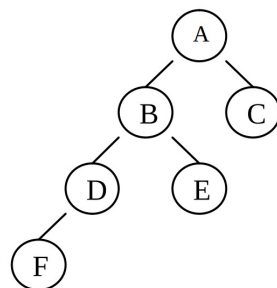


1. Dadas las fuentes de Arboles Binarios proporcionadas por la cátedra, construya un árbol binario de números enteros de por lo menos 5 niveles, y realice la impresión en pantalla de todo sus elementos, realizándolo con los recorridos: preorden, inorden, posorden y por niveles.

2. Agregue a la clase ArbolBinario los siguientes métodos:

a) frontera(): list. Se define frontera de un árbol binario, a las hojas de un árbol binario recorridos de izquierda a derecha.

Ejemplo:



La lista deberá devolver:
F,E,C

b) lleno(): boolean. Devuelve true si el árbol es lleno. Un árbol binario es lleno si tiene todas las hojas en el mismo nivel y además tiene todas las hojas posibles (es decir todos los nodos intermedios tienen dos hijos).

c) completo(): boolean. Devuelve true si el árbol es completo. Un árbol binario de altura h es completo si es lleno hasta el nivel $(h-1)$ y el nivel h se completa de izquierda a derecha.

3. Una red bancaria es una red que posee una topología de árbol binario completo (vea la figura 1 como ejemplo).

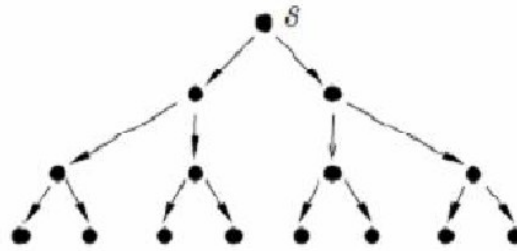


Figura 1: Red binaria completa

Los nodos que conforman una red binaria completa tiene la particularidad de que todos ellos conocen cual es su retardo de reenvío. El retardo de reenvío se define como el período comprendido entre que un nodo recibe un mensaje y lo reenvía a sus dos hijos.

Implementar un algoritmo que calcule el mayor retardo posible, en el camino que realiza un mensaje desde la raíz hasta llegar a las hojas en una red binaria completa.

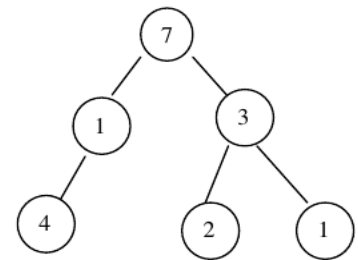
4. Implemente la operación **trayectoriaPesada(ab: ArbolBinario) : list** // Retorna el valor de la trayectoria pesada de cada una de las hojas del árbol binario ab.

Se define el valor de la trayectoria pesada de una hoja de un árbol binario como la suma del contenido de todos los nodos desde la raíz a la hoja multiplicada por el nivel en el que se encuentra. Ejemplo:

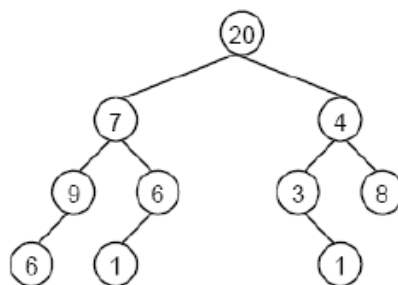
Trayectoria Pesada hoja 4 es $4*2 + 1*1 + 7*0 = 9$

Trayectoria Pesada hoja 2 es $2*2 + 3*1 + 7*0 = 7$

Trayectoria Pesada hoja 1 es $1*2 + 3*1 + 7*0 = 5$



5. Implemente un método que realice un recorrido que llamaremos recorrido_guiado, que permita recorrer un árbol de forma que, en cada iteración, se selecciona el nodo mas pequeño de entre todos los disponibles en ese momento, independientemente de en que rama se encuentre. Se entiende por nodo disponible aquel nodo cuyo padre ya ha sido procesado (excluyendo el nodo raíz). Ejemplo de este recorrido:



Para este árbol, el método **recorrido_guiado** debe imprimir en consola: 20, 4, 3, 1, 7, 6, 1, 8, 9, 6