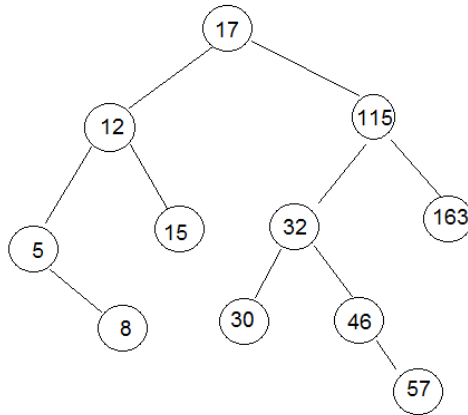


Programación Lógica y Funcional

Práctica 6

En esta práctica haremos uso de **árboles binarios de búsqueda**. Los cuales, como vimos en la clase, tienen las características de no tener nodos repetidos y que todos los descendientes del lado izquierdo de cada nodo son menores que ese nodo y todos los descendientes del lado derecho son mayores. Ejemplo:



Estos árboles son apropiados para realizar búsquedas pues se lleva a cabo de manera muy rápida (logarítmica). Escriba las siguientes funciones. Para las primeras tres funciones puede asumir que el árbol es un árbol binario de búsqueda.

esHoja n a – Esta función regresa **True** si **n** es una hoja del árbol **a**, o **False** si no lo es.

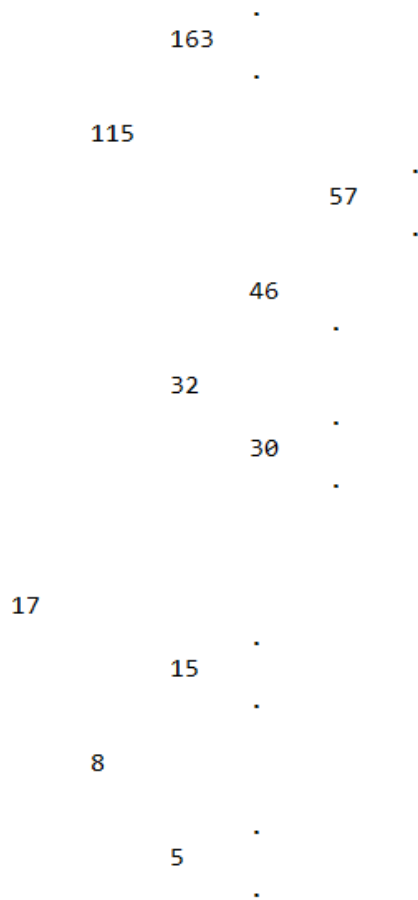
eliminaNodo n a – Regresa un árbol que es el resultado de eliminar el nodo **n** en el árbol **a**.
Restricción: NO DEBEN CONVERTIR EL ÁRBOL A LISTA, LA ELIMINACIÓN DEBE HACERSE DIRECTAMENTE EN EL ÁRBOL

distancia n1 n2 a – Regresa la mínima distancia (cantidad de arcos) que se debe recorrer para ir del nodo **n1** al nodo **n2** en el árbol **a**.

estaOrdenado a – Regresa **True** si el árbol cumple con la restricción de que para cada nodo, todos los sucesores del lado izquierdo son menores y todos los sucesores del lado derecho son mayores, o **False** si no.

Ejemplos:

```
*Main> esHoja 163 t2
True
*Main> esHoja 115 t2
False
*Main> pict (eliminaNodo 12 t2)
```



```
*Main> distancia 5 15 t2
2
*Main> distancia 8 30 t2
6
*Main> estaOrdenado t2
True
*Main> estaOrdenado (Nodo 3 (Nodo 4 Nulo Nulo) Nulo)
False
```