ARBOLES: EJERCICIOS

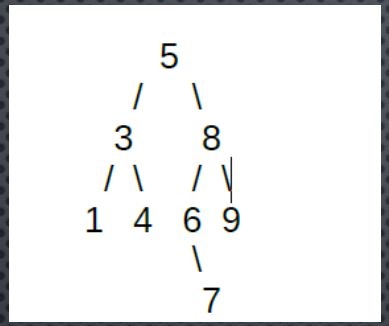
Ejercicio 1.-¿El orden en que las hojas se listan en los recorridos preorden, inorden y postorden de un árbol binario es el mismo en los tres casos?

Ejercicio 2.- Dado un árbol binario cuyas etiquetas están organizadas como un árbol binario de búsqueda, puedo recuperarlo a partir de su preorden.

ARBOLES: EJERCICIOS

Ejercicio3.- Dado un árbol binario de búsqueda, implementa una función para imprimir las etiquetas de los nodos **en orden de mayor a menor profundidad**. Si tienen la misma profundidad pueden aparecer en cualquier orden. Ejemplo:

El resultado seria **7,1,4,6,9,3,8,5.**

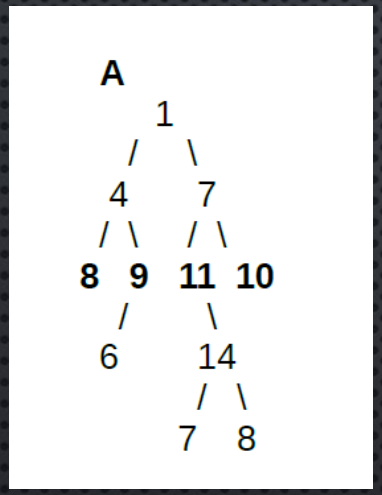


ARBOLES: EJERCICIOS

Ejercicio 4.- Dado un árbol binario cuyas etiquetas están organizadas como un **AVL**, puedo recuperarlo de forma unívoca a partir de su recorrido en inorden

ARBOLES: EJERCICIOS

Ejercicio 5.- Implementa una función **list<int> nivel (const bintree<int> & A);**
que dado un árbol binario A, devuelva una lista con las etiquetas del nivel que tenga un mayor número de nodos



L={8,9,11,10}

ARBOLES: EJERCICIOS

Ejercicio 6.-Construir un **AVL** insertando, en ese orden, las siguientes claves {99, 28, 70, 81, 47, 38, 100, 21, 16, 49, 57}, especificando los pasos seguidos e indicando cuando sea necesario el tipo de rotación que se usa para equilibrar

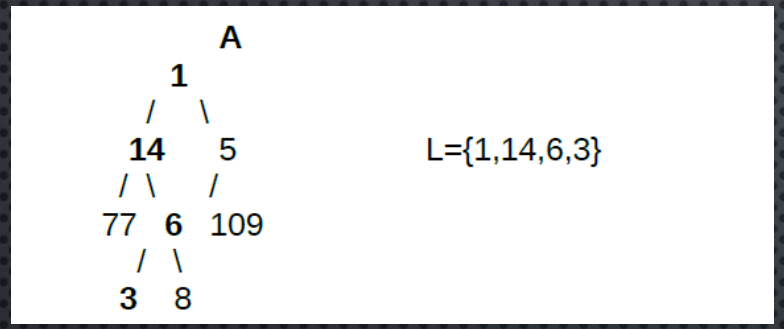
ARBOLES: EJERCICIOS

Ejercicio 7 Construir un **arbol parcialmente ordenado** realizando las siguientes tareas:

- Inserta, en ese orden, las siguientes claves {10, 5, 12, 12, 5, 3, 9, 4, 3}
- Borrar tres elementos.

ARBOLES: EJERCICIOS

Ejercicio 8 Implementar una función **list<int> caminodemenores (const bintree<int> & A);** que dado un árbol binario A, devuelva en L el camino de la raíz a una hoja de forma que la suma de sus etiquetas sea la menor posible. No se pueden usar iteradores



ARBOLES: EJERCICIOS

Ejercicio 9.-Es imposible que un árbol binario (con mas de dos nodos) sea **AVL y APO** a la vez.

Ejercicio 10. El elemento de valor máximo en un **ABB<int>** se encuentra en el nodo más profundo.

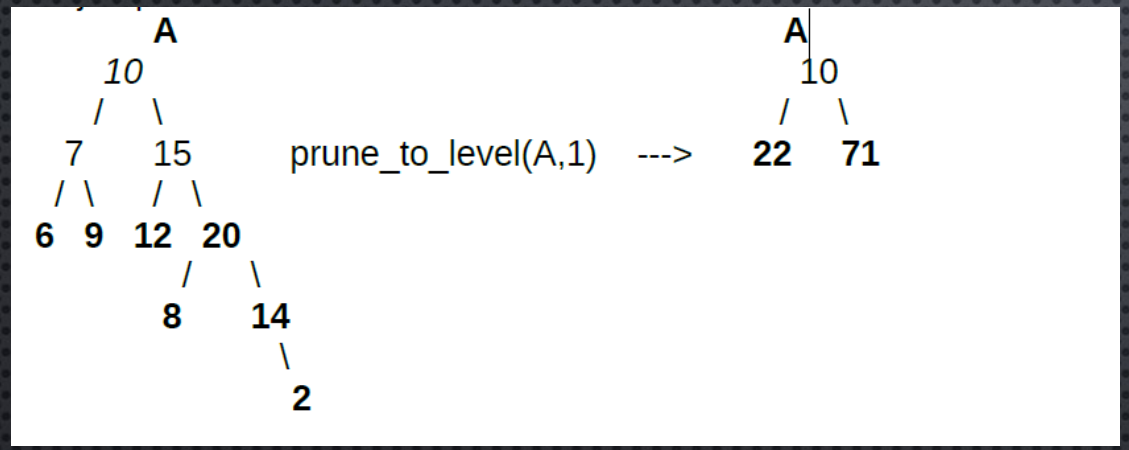
Ejercicio 11. Si inserto las claves {15, 7, 11, 23,18, 19, 9, 30, 1} en un AVL de enteros, **2-a:** Hay que hacer dos rotaciones simples y una rotación doble **2-b:** Hay que hacer tres rotaciones dobles, **2-c:** Hay que hacer dos rotaciones dobles **2-d:** Todo lo anterior es falso

ARBOLES: EJERCICIOS

Ejercicio 12. Dado un árbol binario construir una función que compruebe si es un ABB.

ARBOLES: EJERCICIOS

Ejercicio 13. Implementar una función **void prune_to_level(bintree<int> & A, int level);**
que pode todos los nodos de un árbol binario A por debajo del nivel ***l*evel** sustituyendo la etiqueta de los nodos del nivel más profundo que quede tras podar por la suma de sus descendientes (incluyendo la etiqueta del propio nodo)



Ejercicio 13. continuación

ARBOLES: EJERCICIOS

Ejercicio 14. Dados los siguientes recorridos en **preorden** = (C B F C I H G A J D E), y **postorden** = (F I C B H A D J E G C) **3-a:** No hay ningún árbol binario con esos recorridos asociados; **3-b:** Hay 1 solo árbol binario con esos recorridos asociados: **3-c:** Hay dos árboles binarios con esos recorridos asociados ; **3-d:** Hay múltiples árboles binarios con esos recorridos asociados