

EjercicioABB2

Estructuras de Datos

Tema 4: árboles

1º Grado en Ingeniería de la Computación

© Profesor Dr. Carlos Grima Izquierdo (www.carlosgrima.com)

URJC (www.urjc.es)

Amplía el EjercicioABB1 resuelto para que ahora podamos eliminar del árbol. En concreto, añade lo siguiente:

- Añadiremos un atributo que será la orientación de la siguiente eliminación. Será 1 si podemos por la derecha, o -1 si es por la izquierda. En el constructor, lo inicializamos a 1.
- Añadiremos un método público para eliminar un elemento. Se pasa el elemento a eliminar. Lo buscamos internamente (con el método privado recursivo que hicimos en EjercicioABB1) y, una vez encontrado, lo eliminamos. La precondition será que el elemento a buscar tiene que existir. Este elemento público llamará internamente a un método privado recursivo para eliminar un solo nodo (descrito a continuación).
- Eliminar un nodo (procedimiento recursivo y privado). Le pasamos como único parámetro el puntero al nodo a eliminar. Habrá que considerar cuatro casos:
 1. Si es una hoja, elimina el nodo sin más.
 2. Si no es una hoja y tiene sus dos hijos, pone en su interior el máximo elemento de su subárbol izquierdo, o el mínimo de su subárbol derecho, según lo que indique el atributo de orientación. A continuación cambia el atributo de orientación al valor opuesto, para la próxima eliminación que se haga.
 3. Si no es una hoja y sólo tiene hijo izquierdo, no nos queda más remedio que elegir el máximo de su subárbol izquierdo. A continuación pondremos el atributo de orientación a 1 para que la próxima vez podemos por la derecha.
 4. Si no es una hoja y sólo tiene hijo derecho, no nos queda más remedio que elegir el mínimo de su subárbol derecho. A continuación pondremos el atributo de orientación a -1 para que la próxima vez podemos por la izquierda.En cualquiera de los tres últimos casos, finalmente se llama a sí mismo recursivamente para eliminar el máximo o el mínimo que hemos puesto. La precondition será que el puntero al nodo a eliminar no sea NULL.
- Para calcular el máximo o mínimo de un subárbol, haremos dos sendos métodos privados recursivos, que devolverán el puntero al nodo que contiene el elemento máximo y mínimo respectivamente de un subárbol cuyo puntero al nodo raíz se pasa como único parámetro. Los podríamos hacer iterativos fácilmente (avanzando siempre por la rama de la derecha/izquierda hasta que no podamos más), pero hazlos recursivos para practicar. La precondition es que el puntero a la raíz del subárbol sea distinto de NULL.

Pruébalo con un main en el cual vayamos insertando los mismos nodos que en EjercicioABB1 y en el mismo orden. Una vez insertados todos, imprimiremos el árbol entero. Una vez creado el árbol, eliminaremos el 5, 6, 4 y 7 (imprimiendo después de cada eliminación).

```
Consola de depuración de Microsoft Visual Studio
Arbol inicial:
5
    1 (izquierdo)
        3 (derecho)
            2 (izquierdo)
            4 (derecho)
        6 (derecho)
            7 (derecho)
Eliminamos 5
6
    1 (izquierdo)
        3 (derecho)
            2 (izquierdo)
            4 (derecho)
        7 (derecho)
Eliminamos 6
4
    1 (izquierdo)
        3 (derecho)
            2 (izquierdo)
        7 (derecho)
Eliminamos 4
7
    1 (izquierdo)
        3 (derecho)
            2 (izquierdo)
Eliminamos 7
3
    1 (izquierdo)
        2 (derecho)
```