

Actividad 7.2

Árbol binario de búsqueda

Algoritmos y Estructuras de Datos

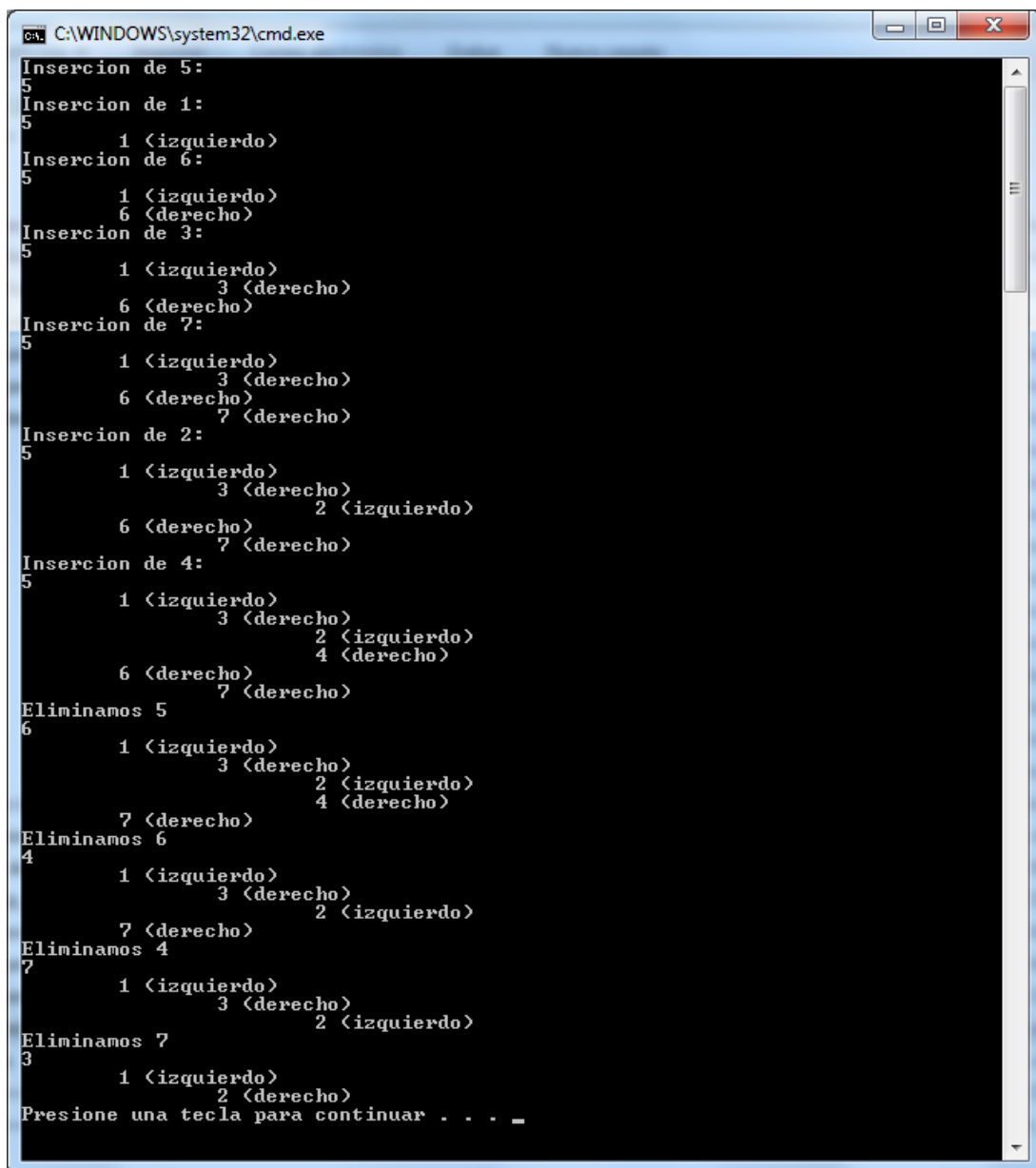
Tema 7: árboles

Programa en C++ la clase “ArbolBinarioDeBusqueda” con al menos las siguientes características:

- Necesitaremos un struct “Nodo” que contenga un int (el contenido) y punteros al padre, hijo izquierdo e hijo derecho.
- La clase “ArbolBinarioDeBusqueda” tendrá, al menos:
 - Atributos:
 - Puntero a la raíz del árbol.
 - Número de nodos del árbol
 - Orientación de la siguiente eliminación. Será 1 si podemos por la derecha, o -1 si es por la izquierda.
 - Métodos:
 - Constructor que construirá un árbol vacío, y destructor que liberará la memoria de todos los nodos. El destructor será recursivo, como en la actividad 7.1.
 - Insertar un elemento: buscará la posición en donde deberíamos insertar el nuevo elemento (para ello usará el método privado descrito a continuación) y lo meterá ahí (sólo como hojas).
 - Un método privado (recursivo) para buscar el hueco en donde se debería insertar un nuevo elemento. Devuelve un puntero al nodo padre de dicho hueco.
 - Buscar un elemento. Buscará la posición del elemento. Si lo encuentra, devuelve un puntero al nodo que lo contiene. Si no, devuelve NULL. Para ello usará el método privado que describimos a continuación.
 - Un método privado (recursivo) para buscar un elemento. Si lo encuentra, devuelve un puntero al nodo que lo contiene. Si no lo encuentra, devuelve NULL.
 - Eliminar un elemento. Se pasa el elemento a eliminar. Lo buscamos (con el método público “buscar”, descrito antes) y, si lo encontramos, lo eliminamos. Usará el procedimiento recursivo descrito a continuación.
 - Eliminar un nodo (procedimiento recursivo). Le pasamos el puntero al nodo a eliminar. Para ello necesitaremos dos métodos recursivos que devolverán el puntero al nodo que contiene el elemento máximo y mínimo respectivamente de un subárbol dado como parámetro. Lo borrado se irá

sustituyendo alternativamente por el mayor del subárbol izquierdo o por el menor del derecho. Para ello, usaremos el atributo dentro de la clase que nos dice si en esta ocasión tenemos que realizar una opción o la otra. Empieza por el subárbol derecho.

- Imprimir. Imprimirá el árbol en forma de esquema numerado, al estilo de anteriores ejercicios sobre árboles. La única diferencia ahora será que, por cada nodo impreso (exceptuando la raíz del árbol completo), entre paréntesis imprimiremos si es hijo izquierdo o hijo derecho de su padre.
- Pruébalo con un main en el cual vayamos insertando hasta conseguir el árbol de la primera figura de la diapositiva 49. Cada vez que insertemos, imprimiremos el árbol entero. La inserción debe hacerse mediante un bucle a partir de un array, al igual que en ejercicios anteriores sobre árboles. Una vez creado el árbol, eliminaremos el 5, 6, 4 y 7 (imprimiendo después de cada eliminación).



```
C:\WINDOWS\system32\cmd.exe
Insercion de 5:
5
Insercion de 1:
5
  1 <izquierdo>
Insercion de 6:
5
  1 <izquierdo>
  6 <derecho>
Insercion de 3:
5
  1 <izquierdo>
    3 <derecho>
  6 <derecho>
Insercion de 7:
5
  1 <izquierdo>
    3 <derecho>
  6 <derecho>
    7 <derecho>
Insercion de 2:
5
  1 <izquierdo>
    3 <derecho>
      2 <izquierdo>
  6 <derecho>
    7 <derecho>
Insercion de 4:
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>
Presione una tecla para continuar . . . _
```