

Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías



IL365 - Estructura de Datos - D01

Actividad de Aprendizaje #13

El Árbol AVL, Implementación Dinámica

Alumna: Cervantes Araujo Maria Dolores

Código: 217782452

Fecha de Elaboración: 28 abril de 2023



Autoevaluación			
Concepto	Si	No	Acumulación
Bajé el trabajo de internet o alguien me lo pasó (aunque sea de forma parcial)	-100 pts	0 pts	0
Incluí el código fuente en formato de texto (sólo si funciona cumpliendo todos los requerimientos)	+25pts	0 pts	25
Incluí las impresiones de pantalla (sólo si funciona cumpliendo todos los requerimientos)	+25pts	0 pts	25
Incluí una portada que identifica mi trabajo (nombre, código, materia, fecha, título)	+25 pts	0 pts	25
Incluí una descripción y conclusiones de mi trabajo	+25 pts	0 pts	25
Suma:			100

Introducción:

Un árbol AVL (Adelson–Velskii y Landis) es un árbol binario de búsqueda que satisface la condición de estar balanceado. La semana pasada vimos un Árbol Binario de Búsqueda, el cual respeta la propiedad de orden en todos sus nodos, es decir, todas las claves en su subárbol izquierdo son menores que la clave del nodo y todas las claves en el subárbol derecho son mayores.

En esta ocasión vemos el árbol AVL, contiene la propiedad de balanceo, la cual nos dice que, para cada nodo del árbol, la diferencia de altura entre el subárbol izquierdo y el subárbol derecho no debe de ser mayor a 2 o -2.

Nos guiamos de la actividad pasada, para de ahí partir a realizar el árbol AVL, solo modificamos algunos aspectos del modelo como quitar algunas líneas de código y agregar un detalle en la función de insertar, también le agregamos las funciones para el balanceo tanto simple como doble, del lado izquierdo y derecho.



Código Fuente:

[arbolAVL.hpp](#)

```
#ifndef ARBOLAVL_HPP_INCLUDED
#define ARBOLAVL_HPP_INCLUDED
#include <iostream>
#include <string>
#include <exception>

template <class T>
class ArbolAVL {
private:
    class Node {
private:
        T* dataPtr;
        Node* left;
        Node* right;

public:
        class Exception: public std::exception {
private:
            std::string msg;
public:
            explicit Exception(const char* message): msg(message) {}
            explicit Exception(const std::string& message):
                msg(message) {}

            virtual ~Exception() throw() {}
            virtual const char* what() const throw() {
                return msg.c_str();
            }
        };

        Node();
        Node(const T&);
        ~Node();

        T* getDataPtr() const;
        T getData() const;
        Node*& getLeft();
        Node*& getRight();

        void setDataPtr(T*);
        void setData(const T&);
        void setLeft(Node*&);
        void setRight(Node*&);
    };

    Node* root;

public:
    void copyAll(const ArbolAVL<T>&);
    void insertData(Node*&, const T&);
};
```



```
void deleteData (Node*&, const T&);

Node*& findData (Node*&, const T&);
Node*& getLowest (Node*&);
Node*& getHighest (Node*&);

void preOrder (Node*&);
void inOrder (Node*&);
void posOrder (Node*&);

int getBalanceFactor (Node*&);
void simpleLeftRoot (Node*&);
void simpleRightRoot (Node*&);
void doubleLeftRoot (Node*&);
void doubleRightRoot (Node*&);
void balancing (Node*&);

public:
class Exception: public std::exception {
private:
    std::string msg;
public:
    explicit Exception(const char* message): msg(message) {}
    explicit Exception(const std::string& message): msg(message) {}
    virtual ~Exception() throw() {}
    virtual const char* what() const throw() {
        return msg.c_str();
    }
};

ArbolAVL();
ArbolAVL(const ArbolAVL<T>&);
~ArbolAVL();

bool isEmpty() const;
bool isLeaf(Node*&) const;

void insertData(const T&);
void deleteData(const T&);
void deleteAll();
T retrieve(Node*&) const;

int getHeight (Node*&);
int getHeight ();
int getHeightRight ();
int getHeightLeft ();
Node*& findData (const T&);
Node*& getLowest ();
Node*& getHighest ();

void preOrder ();
void inOrder ();
void posOrder ();
```



```
        ArbolAVL& operator = (const ArbolAVL&);
    };

#endif // ARBOLAVL_HPP_INCLUDED

///IMPLEMENTACION NODO

using namespace std;
template <class T>
ArbolAVL<T>::Node::Node() : dataPtr(nullptr), left(nullptr), right(nullptr) { }

template <class T>
ArbolAVL<T>::Node::Node(const T& m) : dataPtr(new T(m)), left(nullptr),
right(nullptr) {
    if(dataPtr == nullptr) {
        throw Exception("Memoria insuficiente, se creara un nodo");
    }
}

template <class T>
ArbolAVL<T>::Node::~~Node() {
    delete dataPtr;
}

template <class T>
T* ArbolAVL<T>::Node::getDataPtr() const {
    return dataPtr;
}

template <class T>
T ArbolAVL<T>::Node::getData() const {
    if(dataPtr == nullptr) {
        throw Exception("Dato inexistente, getData");
    }
    return *dataPtr;
}

template <class T>
typename ArbolAVL<T>::Node*& ArbolAVL<T>::Node::getLeft() {
    return left;
}

template <class T>
typename ArbolAVL<T>::Node*& ArbolAVL<T>::Node::getRight() {
    return right;
}

template <class T>
void ArbolAVL<T>::Node::setDataPtr(T* pos) {
    dataPtr = pos;
}
```



```
template <class T>
void ArbolAVL<T>::Node::setData(const T& e) {
    if(dataPtr == nullptr) {
        if((dataPtr = new T(e)) == nullptr) {
            throw Exception("Memoria no disponible, setData");
        }
    }
    else {
        *dataPtr = e;
    }
}

template <class T>
void ArbolAVL<T>::Node::setLeft(Node*& pos) {
    left = pos;
}

template <class T>
void ArbolAVL<T>::Node::setRight(Node*& pos) {
    right = pos;
}

///IMPLEMENTACION ARBOL

template <class T>
ArbolAVL<T>::ArbolAVL() : root(nullptr) {}

template <class T>
ArbolAVL<T>::ArbolAVL(const ArbolAVL& t) : root(nullptr) {
    copyAll(t);
}

template <class T>
ArbolAVL<T>::~~ArbolAVL() {
    deleteAll();
}

template <class T>
void ArbolAVL<T>::copyAll(const ArbolAVL<T>& l) {
    Node* newNode;

    try {
        if((newNode = new Node(l.aux->getData())) == nullptr) {
            throw Exception("Memoria no disponible, copyAll");
        }
    }
    catch(Exception ex) {
        throw Exception(ex.what());
    }

    root = new Node(l.root->getData());
    root->getLeft() = l.root->getLeft();
    root->getRight() = l.root->getRight();
}
```



```
    }

template <class T>
bool ArbolAVL<T>::isEmpty() const {
    return root == nullptr;
}

template <class T>
void ArbolAVL<T>::insertData(const T& d) {
    insertData(root, d);
}

template <class T>
void ArbolAVL<T>::insertData(Node*& r, const T& d) {
    if(r == nullptr) { //esta balanceado, inserta como hoja
        try {
            if((r= new Node(d)) == nullptr) {
                throw Exception("Memoria no disponible, insertData");
            }
        }
        catch(typename Node::Exception ex) {
            throw Exception(ex.what());
        }
    }
    //inserta recursivamente
    else {
        if(d < r->getData()) {
            insertData(r->getLeft(), d);
        }
        else {
            insertData(r->getRight(), d);
        }
        balancing(r);
    }
}

template <class T>
int ArbolAVL<T>::getBalanceFactor(Node*& e) {
    return (e == nullptr) ? 0 : getHeight(e->getRight()) - getHeight(e->getLeft());
}

template <class T>
void ArbolAVL<T>::simpleLeftRoot(Node*& e) {
    Node* aux1(e->getRight());
    Node* aux2(aux1->getLeft());

    e->setRight(aux2);
    aux1->setLeft(e);
    e = aux1;
}

template <class T>
void ArbolAVL<T>::simpleRightRoot(Node*& e) {
```



```
Node* aux1(e->getLeft());  
Node* aux2(aux1->getRight());
```

```
e->setLeft(aux2);  
aux1->setRight(e);  
e = aux1;  
}
```

```
template <class T>  
void ArbolAVL<T>::doubleLeftRoot(Node*& e) {  
    simpleRightRoot(e->getRight());  
    simpleLeftRoot(e);  
}
```

```
template <class T>  
void ArbolAVL<T>::doubleRightRoot(Node*& e) {  
    simpleLeftRoot(e->getLeft());  
    simpleRightRoot(e);  
}
```

```
template <class T>  
void ArbolAVL<T>::balancing(Node*& e) {  
    switch(getBalanceFactor(e)) {  
        ///Desbalanceado a la izq  
        case -2:  
            getBalanceFactor(e->getLeft()) == -1 ? simpleRightRoot(e) :  
doubleRightRoot(e);  
            break;  
        ///desbalanceado a la der  
        case 2:  
            getBalanceFactor(e->getRight()) == 1 ? simpleLeftRoot(e) :  
doubleLeftRoot(e);  
            break;  
    }  
}
```

```
template <class T>  
void ArbolAVL<T>::deleteData(const T& d) {  
    deleteData(root, d);  
}
```

```
template <class T>  
void ArbolAVL<T>::deleteData(Node*& e, const T& d) {  
    if(e == nullptr) {  
        return;  
    }  
  
    if(e->getData() == d) {  
        if(isLeaf(e)) { ///si es hoja  
            delete e;  
            e = nullptr;  
            return;  
        }  
    }  
}
```




```
///caso de dos hojas
if(e->getLeft() != nullptr && e->getRight() != nullptr) {
    T myData(getHighest(e->getLeft())->getData());

    deleteData(e->getLeft(), myData);
    e->setData(myData);
    return;
}

///Casi una sola hoja
Node* aux(e);
e = e->getLeft() == nullptr ? e->getRight() : e->getLeft();
delete aux;
return;
}

if(d < e->getData()) {
    deleteData(e->getLeft(), d);
}
else {
    deleteData(e->getRight(), d);
}
}

template <class T>
T ArbolAVL<T>::retrieve(Node*& r) const {
    return r->getData();
}

template <class T>
typename ArbolAVL<T>::Node*& ArbolAVL<T>::findData(const T& e) {
    return findData(root, e);
}

template <class T>
typename ArbolAVL<T>::Node*& ArbolAVL<T>::findData(Node*& r, const T& e) {
    if(r == nullptr || r->getData() == e) {
        return r;
    }

    if(e < r->getData()) {
        return findData(r->getLeft(), e);
    }
    else {
        return findData(r->getRight(), e);
    }
}

template <class T>
typename ArbolAVL<T>::Node*& ArbolAVL<T>::getLowest() {
    return getLowest(root);
}
```



```
template <class T>
typename ArbolAVL<T>::Node*& ArbolAVL<T>::getLowest(Node*& r) {
    if(r == nullptr || r->getLeft() == nullptr) {
        return r;
    }
    return getLowest(r->getLeft());
}

template <class T>
typename ArbolAVL<T>::Node*& ArbolAVL<T>::getHighest() {
    return getHighest(root);
}

template <class T>
typename ArbolAVL<T>::Node*& ArbolAVL<T>::getHighest(Node*& r) {
    if(r == nullptr || r->getRight() == nullptr) {
        return r;
    }
    return getHighest(r->getRight());
}

template <class T>
bool ArbolAVL<T>::isLeaf(Node*& r) const {
    return r != nullptr && r->getLeft() == r->getRight();
}

template <class T>
int ArbolAVL<T>::getHeight() {
    return getHeight(root);
}

template <class T>
int ArbolAVL<T>::getHeight(Node*& r) {
    if(r == nullptr) {
        return 0;
    }

    int LH(getHeight(r->getLeft()));
    int rH(getHeight(r->getRight()));

    return (LH > rH ? LH : rH) + 1;
}

template <class T>
int ArbolAVL<T>::getHeightRight() {
    return getHeight(root->getRight());
}

template <class T>
int ArbolAVL<T>::getHeightLeft() {
    return getHeight(root->getLeft());
}
```



```
template <class T>
void ArbolAVL<T>::preOrder() {
    preOrder(root);
}

template <class T>
void ArbolAVL<T>::preOrder(Node*& r) {
    if(r == nullptr) {
        return;
    }
    std::cout << r->getData() << ", ";
    preOrder(r->getLeft());
    preOrder(r->getRight());
}

template <class T>
void ArbolAVL<T>::inOrder() {
    inOrder(root);
}

template <class T>
void ArbolAVL<T>::inOrder(Node*& r) {
    if(r == nullptr) {
        return;
    }
    inOrder(r->getLeft());
    std::cout << r->getData() << ", ";
    inOrder(r->getRight());
}

template <class T>
void ArbolAVL<T>::posOrder() {
    posOrder(root);
}

template <class T>
void ArbolAVL<T>::posOrder(Node*& r) {
    if(r == nullptr) {
        return;
    }
    posOrder(r->getLeft());
    posOrder(r->getRight());
    std::cout << r->getData() << ", ";
}

template <class T>
void ArbolAVL<T>::deleteAll() {
    if(root == nullptr) {
        return;
    }
    deleteData(root->getLeft(), root->getData());
    deleteData(root->getRight(), root->getData());
}
```



```
delete root;  
root = nullptr;  
}
```

```
template <class T>  
ArbolAVL<T>& ArbolAVL<T>::operator = (const ArbolAVL& e) {  
    deleteAll();  
    copyAll(e);  
    return *this;  
}
```

menu.hpp

```
#ifndef MENU_HPP_INCLUDED  
#define MENU_HPP_INCLUDED  
#include <string>  
#include <random>  
#include <chrono>  
#include <iostream>  
#include <windows.h>  
#include <functional>  
#include "arbolAVL.hpp"  
  
class Menu {  
    private:  
        void view(ArbolAVL<int>&);  
    public:  
        Menu(ArbolAVL<int>&);  
};  
#endif // MENU_HPP_INCLUDED
```

menu.cpp

```
#include "menu.hpp"  
  
using namespace std;  
  
Menu::Menu(ArbolAVL<int> &tree) {  
    system("Color E5");  
    view(tree);  
}  
  
void Menu::view(ArbolAVL<int> &tree) {  
    int num, i=0, val;  
    std::chrono::steady_clock::time_point begin, end;  
    default_random_engine  
generator(chrono::system_clock::now().time_since_epoch().count());  
    uniform_int_distribution<int>distribution(0,100000);  
    auto random = bind(distribution, generator);  
  
    cout<<"Cuantos numeros desea insertar en el arbol?"<<endl;
```



```
cin>>num;
cin.ignore();
cout<<endl<<endl;

while(i < num) {
    val = random();
    cout<<"Insertando: "<< val << endl;
    tree.insertData(val);
    i++;
}

cout << endl << endl << endl;

cout << "----- RECORRIDO PRE-ORDER -----" << endl << endl;
tree.preOrder();
cout << endl << endl;

cout << "----- RECORRIDO IN-ORDER -----" << endl << endl;
tree.inOrder();
cout << endl << endl;

cout << "----- RECORRIDO POST-ORDER -----" << endl << endl;
tree.posOrder();
cout << endl << endl;

cout << "----- ELEMENTO MENOR EN EL ARBOL -----" << endl << endl;
cout<< "- " << tree.retrieve(tree.getLowest());
cout << endl << endl;

cout << "----- ELEMENTO MAYOR EN EL ARBOL -----" << endl << endl;
cout<< "- " << tree.retrieve(tree.getHighest());
cout << endl << endl;

cout << "----- ALTURA DEL ARBOL -----" << endl << endl;
cout<< "- " << tree.getHeight();
cout << endl << endl;

cout << "----- ALTURA DEL SUB-ARBOL IZQUIERDO -----" << endl <<
endl;
cout<< "- " << tree.getHeightLeft();
cout << endl << endl;

cout << "----- ALTURA DEL SUB-ARBOL DERECHO -----" << endl <<
endl;
cout<< "- " << tree.getHeightRight();
cout << endl << endl;

cout << "Eliminando arbol..." << endl << endl;
tree.deleteAll();
cout << "Se elimino correctamente :)"<<endl;
}
```



main.cpp

```
#include "menu.hpp"
int main() {
    ArbolAVL<int> m;
    Menu start(m);
}
```

Impresiones de Pantalla:

Sin AVL:

```
"C:\Users\cervel\Escritorio\F.Prog\Estructura de datos\arbol\AVL\bin\Debug\arbolAVL.exe"
Cuántos números desea insertar en el árbol?
1000

Insertando: 31443
Insertando: 47743
Insertando: 98689
Insertando: 15286
Insertando: 10286
Insertando: 86813
Insertando: 9499
Insertando: 48662
Insertando: 45304
Insertando: 7153
Insertando: 28428
Insertando: 73864
Insertando: 145
Insertando: 37020
Insertando: 73086
Insertando: 16615
Insertando: 46520
Insertando: 45883
Insertando: 37611
Insertando: 16047
Insertando: 97184
Insertando: 18910
Insertando: 8218
Insertando: 24501
Insertando: 78669
Insertando: 42880
Insertando: 57382
Insertando: 83308
Insertando: 12512
Insertando: 53652
Insertando: 4332
Insertando: 12460
Insertando: 10235
Insertando: 20648
Insertando: 33103
Insertando: 45323

8, 97184, 86813, 99061, 99374, 99300, 99500, 99702, 99624, 99583, 99843, 99820, 99284, 98689, 47743, 31443,
----- ELEMENTO MENOR EN EL ARBOL -----
- 34
----- ELEMENTO MAYOR EN EL ARBOL -----
- 99843
----- ALTURA DEL ARBOL -----
- 22
----- ALTURA DEL SUB-ARBOL IZQUIERDO -----
- 19
----- ALTURA DEL SUB-ARBOL DERECHO -----
- 21
Eliminando árbol...
Se eliminó correctamente :)
```



Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
ICOM – Ingeniería en Computación
Módulo Estructura de Datos



Prueba 1 con AVL:

"C:\Users\cerva\Escritorio\F.Prog\Estructura de datos\arbol\AVL\bin\Debug\arbolAVL.exe"

Cuántos números desea insertar en el árbol?
1800

Insertando: 54022
Insertando: 19175
Insertando: 67802
Insertando: 16098
Insertando: 49774
Insertando: 21936
Insertando: 67491
Insertando: 90170
Insertando: 35806
Insertando: 84409
Insertando: 17895
Insertando: 51492
Insertando: 8179
Insertando: 8756
Insertando: 58793
Insertando: 49026
Insertando: 62401
Insertando: 47598
Insertando: 52741
Insertando: 83896
Insertando: 99124
Insertando: 12220
Insertando: 82755
Insertando: 22411
Insertando: 60760
Insertando: 61105
Insertando: 68180
Insertando: 67998
Insertando: 76477
Insertando: 8155
Insertando: 64949
Insertando: 19372
Insertando: 74183
Insertando: 51381
Insertando: 36743
Insertando: 29624
Insertando: 84150
Insertando: 55051
Insertando: 19571
Insertando: 21088
Insertando: 28791
Insertando: 74755
Insertando: 65292
Insertando: 29391
Insertando: 56289
Insertando: 14472

"C:\Users\cerva\Escritorio\F.Prog\Estructura de datos\arbol\AVL\bin\Debug\arbolAVL.exe"

----- RECORRIDO PRE-ORDER -----

49774, 22934, 12220, 6097, 3558, 2860, 1225, 667, 365, 67, 544, 534, 789, 755, 1035, 1591, 1329, 1927, 1725, 1973, 2786, 2484, 2372, 2113, 2644, 2715, 3239, 2971, 2886, 2866, 2899, 3093, 3186, 3
322, 3261, 3257, 3301, 3514, 3431, 4678, 4118, 3812, 3686, 3907, 3903, 4510, 4347, 4324, 4503, 4630, 5339, 5041, 4786, 4863, 5213, 5370, 6002, 8756, 7474, 6505, 6243, 6241, 5208, 6391, 6291, 648
2, 7133, 7037, 6789, 6690, 6872, 7119, 7158, 7155, 7223, 7385, 8155, 7605, 7539, 7773, 7736, 7904, 8326, 8170, 8426, 8375, 8572, 10539, 9847, 9573, 9182, 8810, 9038, 9370, 9204, 9487, 9744, 9676
9752, 10115, 10114, 9883, 10204, 10293, 11379, 11023, 10824, 10794, 10793, 10942, 11218, 11098, 12067, 11788, 11580, 11399, 11902, 11992, 12134, 12094, 19175, 16098, 14472, 13926, 12479, 12221
12220, 12294, 12941, 12853, 13016, 14249, 14068, 14109, 14402, 14470, 15664, 15122, 14751, 14702, 14643, 14711, 14886, 14908, 15357, 15661, 15856, 15779, 15774, 15887, 15979, 16034, 18113, 178
95, 17295, 16805, 16194, 16993, 17111, 17391, 17389, 17850, 17876, 17943, 17896, 17980, 18065, 17996, 18884, 18746, 18535, 18237, 18229, 18326, 18318, 18737, 18612, 18984, 18891, 18864, 1
8965, 19107, 21088, 20034, 19571, 19372, 19248, 19200, 19256, 19539, 19411, 19569, 19702, 19607, 19719, 20444, 20186, 20159, 20054, 20299, 20254, 20436, 20674, 20445, 20817, 21936, 21414, 21220
21174, 21113, 21187, 21358, 21392, 21788, 21469, 21442, 21616, 21839, 21777, 22411, 22091, 22017, 21990, 22024, 22362, 22788, 22634, 22853, 35806, 28791, 25199, 24583, 23719, 23539, 23333, 2310
9, 23446, 23350, 23633, 23548, 23688, 24055, 23852, 23770, 24207, 24077, 24245, 24796, 24745, 24712, 24833, 24819, 25167, 24857, 25181, 26860, 25790, 25710, 25443, 25351, 25363, 25500, 25723, 25
756, 26053, 25916, 25849, 25975, 26610, 26549, 26107, 26657, 28015, 27440, 27243, 26980, 27381, 27617, 27525, 27617, 28350, 28256, 28048, 28495, 28455, 33068, 30832, 29791, 29393, 29038, 28933,
29993, 29624, 29404, 29661, 29747, 30392, 29808, 29870, 30259, 30215, 30311, 30670, 30651, 32002, 31726, 31136, 30979, 31424, 32010, 32462, 32229, 32205, 32369, 32445, 32746, 32559, 32778, 34078
33271, 33117, 33094, 33008, 33095, 33159, 34002, 33470, 33438, 33441, 33402, 33483, 34003, 34008, 34593, 34455, 34306, 34175, 34386, 34508, 34490, 34562, 35073, 34721, 34720, 34886, 35323, 351
78, 35190, 35612, 43824, 40265, 38152, 37052, 36743, 36659, 36249, 36230, 36574, 36732, 36951, 36903, 36982, 37491, 37293, 37287, 37451, 37384, 37852, 37749, 38081, 38095, 38469, 38345, 38203, 3
8152, 38310, 38369, 38557, 38545, 38603, 38975, 38853, 38715, 38618, 38950, 38856, 40046, 39329, 39473, 40155, 41953, 41090, 40888, 40430, 40347, 40578, 40458, 40649, 40941, 40990, 41045, 41374,
41208, 41107, 41281, 41681, 41513, 41746, 42214, 42065, 42060, 42025, 42177, 42132, 42209, 43136, 42597, 42555, 42360, 42594, 42917, 42627, 43092, 43420, 43161, 43611, 47648, 46342, 45175, 4445
0, 44298, 44132, 43913, 44138, 44436, 44822, 44592, 44549, 44948, 45873, 45704, 45248, 45733, 46178, 45873, 45885, 46260, 47343, 47125, 46789, 46784, 47110, 47009, 47295, 47144, 47312, 47327, 47
578, 47500, 47466, 47598, 48068, 48105, 47907, 47854, 47678, 47886, 47923, 48007, 48426, 48191, 48183, 48392, 48525, 48445, 48567, 49026, 48781, 48680, 48682, 48880, 48801, 48999, 49223, 49060,
49032, 49617, 49422, 49617, 82755, 67491, 57694, 54022, 51492, 50472, 49925, 49851, 49896, 50103, 49956, 49968, 50435, 51247, 50834, 50496, 50526, 50857, 50941, 51407, 51381, 51407, 52355, 52014
51716, 51518, 51764, 52112, 52134, 53200, 52741, 52731, 52643, 52874, 52850, 53178, 53590, 53204, 54010, 56289, 55051, 54720, 54415, 54171, 54068, 54478, 54422, 54733, 54731, 54904, 54814, 550
50, 55847, 55227, 55178, 55122, 55508, 55377, 55556, 56180, 56173, 56256, 57520, 57062, 56587, 56775, 57355, 57259, 57477, 57588, 57566, 57558, 57614, 57668, 64119, 61105, 59785, 58635, 58302, 5
7906, 57793, 57996, 58448, 58335, 58495, 58585, 59290, 59023, 58672, 59142, 59467, 59309, 59580, 60250, 60046, 59808, 59789, 59901, 60078, 60072, 60161, 60116, 60178, 60760, 60430, 60415, 60325,
60587, 60433, 60725, 60786, 60906, 62401, 61658, 61560, 61350, 61264, 61630, 62126, 61981, 61757, 62025, 62041, 62368, 62248, 63509, 62935, 62456, 62450, 62477, 63107, 63064, 63838, 63820, 6380
5, 63972, 64090, 65522, 65068, 64940, 64371, 64328, 64834, 64468, 64843, 65009, 64967, 65011, 65210, 65182, 65080, 65187, 65292, 65464, 66574, 66028, 65848, 65668, 65807, 65936, 66216, 66089, 66
097, 66542, 66301, 66560, 67098, 66931, 67316, 67178, 67408, 74755, 71999, 68703, 68063, 67990, 67802, 67571, 67628, 67828, 67935, 68005, 68084, 68263, 68122, 68112, 68180, 68598, 70213, 69038,
68777, 68713, 68713, 68860, 68839, 68965, 69556, 69486, 69471, 69510, 69899, 69737, 71271, 70403, 70274, 70779, 70882, 71725, 71482, 71902, 73685, 72724, 72327, 72126, 72053, 72633, 72549, 72661
73216, 72811, 72783, 72741, 73137, 73578, 73451, 73553, 73482, 73578, 73507, 73628, 73588, 73780, 73744, 73725, 73918, 74247, 74129, 74183, 74595, 74248, 74668, 76291, 77278, 75971, 75256, 748
08, 74781, 75014, 74963, 75720, 75683, 75895, 75824, 75952, 76477, 76146, 76041, 76023, 76072, 76367, 76228, 76450, 76850, 76564, 76954, 77043, 77527, 77425, 77283, 77532, 77747, 77737, 78230, 7
8038, 77894, 78158, 78100, 78141, 78217, 80958, 79445, 78748, 78542, 78291, 78365, 78646, 78710, 79253, 78996, 78938, 79203, 79363, 79288, 79417, 80391, 79960, 79870, 79522, 79806, 80164, 80017,
80383, 80712, 80651, 80608, 80719, 81899, 81728, 81449, 81117, 81037, 81705, 81840, 81756, 81748, 81887, 82224, 82149, 82008, 82196, 82213, 82358, 82360, 82346, 82625, 82527, 82715, 90170, 8610
1, 84409, 83896, 83190, 82789, 82761, 83065, 83342, 83340, 83219, 83381, 83354, 83434, 84150, 83905, 84262, 84214, 85374, 85109, 85059, 84771, 85072, 85259, 85206, 85462, 85393, 85955, 85918, 86
082, 87319, 86884, 85568, 86262, 86142, 86161, 85906, 86409, 85532, 85702, 86638, 86747, 86725, 86801, 87208, 87107, 86943, 86916, 86945, 87120, 87111, 87144, 87242, 87316, 88694, 87795, 87690,
87584, 87867, 87864, 87837, 88216, 88081, 89618, 89311, 88878, 88827, 89295, 89459, 89411, 89594, 89733, 89676, 89640, 89988, 89986, 90061, 94767, 93116, 92294, 91308, 90416, 90381, 90234, 90951
90483, 91277, 91642, 91496, 91404, 92021, 91711, 92540, 92333, 92297, 92329, 92449, 92369, 92834, 92979, 93878, 93459, 93168, 93250, 93690, 93802, 94228, 94065, 93923, 94097, 94462, 94322, 942
83, 94581, 94606, 97490, 96594, 95881, 95516, 95248, 95015, 94842, 95224, 95366, 95478, 95559, 95528, 95577, 95565, 95594, 96467, 96033, 95942, 95920, 96395, 96277, 96535, 96570, 97190, 97007, 9
6849, 96819, 96784, 96993, 96884, 97128, 97068, 97170, 97179, 97360, 97331, 97235, 97465, 97468, 98355, 97881, 97758, 97522, 97595, 97877, 98029, 97980, 97956, 98189, 98081, 98296, 99124, 98814,
98659, 98593, 98775, 98885, 99462, 99319, 99173, 99178, 99327, 99385, 99862, 99574, 99785, 99944, 99924,

[illegible]



Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
ICOM – Ingeniería en Computación
Módulo Estructura de Datos



```
"C:\Users\cervat\Escritorio\F.Prog\Estructura de datos\arbol\AVL\bin\Debug\arbolAVL.exe"
, 91308, 92329, 92297, 92369, 92449, 92333, 92979, 92834, 92540, 92294, 93250, 93168, 93802, 93690, 93459, 93923, 94097, 94065, 94283, 94322, 94606, 94581, 94461, 94228, 93878, 93116, 94842, 952
24, 95015, 95478, 95366, 95248, 95528, 95565, 95594, 95577, 95559, 95516, 95920, 95942, 96277, 96395, 96033, 96570, 96535, 96467, 95881, 96784, 96819, 96884, 96993, 96849, 97068, 97179, 97170, 9
7128, 97007, 97235, 97331, 97468, 97465, 97360, 97190, 96594, 97595, 97522, 97877, 97758, 97956, 97980, 98081, 98296, 98189, 98019, 97881, 98593, 98775, 98659, 98885, 98814, 99178, 99173, 99385,
99327, 99319, 99785, 99574, 99924, 99944, 99862, 99462, 99124, 98355, 97490, 94767, 90170, 82755, 49774,

----- ELEMENTO MENOR EN EL ARBOL -----
- 67
----- ELEMENTO MAYOR EN EL ARBOL -----
- 99944
----- ALTURA DEL ARBOL -----
- 12
----- ALTURA DEL SUB-ARBOL IZQUIERDO -----
- 10
----- ALTURA DEL SUB-ARBOL DERECHO -----
- 11
Eliminando arbol...
Se elimino correctamente :)
Process returned 0 (0x0)   execution time : 5.770 s
Press any key to continue.
```

Prueba 2 con AVL:

```
"C:\Users\cervat\Escritorio\F.Prog\Estructura de datos\arbol\AVL\bin\Debug\arbolAVL.exe"
Cuantos numeros desea insertar en el arbol?
1200

Insertando: 51073
Insertando: 54971
Insertando: 65013
Insertando: 47408
Insertando: 55366
Insertando: 6924
Insertando: 83080
Insertando: 83572
Insertando: 48179
Insertando: 21076
Insertando: 14408
Insertando: 52208
Insertando: 97626
Insertando: 47383
Insertando: 36298
Insertando: 18752
Insertando: 62695
Insertando: 74283
Insertando: 27308
Insertando: 23005
Insertando: 39867
Insertando: 22637
Insertando: 52629
Insertando: 14229
Insertando: 42296
Insertando: 89988
Insertando: 76789
Insertando: 58766
Insertando: 43860
Insertando: 31693
Insertando: 39372
Insertando: 11799
Insertando: 86487
Insertando: 46359
Insertando: 29567
Insertando: 14943
Insertando: 38981
Insertando: 41000
Insertando: 68456
Insertando: 39814
Insertando: 41349
Insertando: 39995
Insertando: 76825
Insertando: 44193
Insertando: 35217
Insertando: 77613
```




Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
ICOM – Ingeniería en Computación
Módulo Estructura de Datos



"C:\Users\cervat\Escritorio\F.Prog\Estructura de datos\arbol\AVL\bin\Debug\arbolAVL.exe"

RECORRIDO PRE-ORDER

47908, 21076, 9991, 4958, 2688, 654, 218, 102, 46, 76, 157, 128, 387, 351, 353, 538, 537, 582, 1698, 1428, 1001, 995, 999, 1376, 1174, 1661, 1546, 1545, 1577, 1670, 1662, 2113, 1279, 2045, 2199, 2171, 2226, 2614, 3648, 3200, 2738, 2696, 2919, 2811, 3138, 3271, 3251, 3202, 3522, 3368, 4466, 4111, 3732, 3680, 4004, 4336, 4187, 4767, 4563, 4585, 4846, 4384, 5656, 5144, 5067, 4990, 5083, 5 073, 5162, 5241, 5226, 5400, 5372, 5538, 6198, 6042, 5908, 5789, 6128, 6055, 6348, 6225, 6383, 8404, 7794, 6924, 6873, 6779, 6652, 6876, 7337, 7052, 7561, 7689, 8097, 7974, 7842, 8056, 8161, 815 6, 8121, 8195, 9280, 8751, 8701, 8646, 8927, 8871, 8765, 9258, 9240, 9628, 9458, 9608, 9836, 9760, 9884, 24408, 12883, 11709, 11005, 10595, 10323, 10041, 9950, 10107, 10339, 10543, 10675, 10796, 11278, 11249, 11060, 11014, 11202, 11259, 11321, 11288, 11281, 11296, 11421, 11335, 11509, 12146, 12083, 11867, 11793, 11872, 11922, 12124, 12034, 12439, 12167, 12835, 12627, 12840, 13591, 1307 0, 13038, 12947, 12924, 13054, 13523, 13171, 14116, 13790, 13777, 13674, 13753, 13787, 14066, 14022, 14107, 14293, 14229, 14198, 14261, 14346, 14314, 17709, 15592, 15073, 14551, 14450, 14427, 14 680, 14673, 14551, 14676, 14943, 14858, 15017, 15206, 15161, 15107, 15493, 15268, 15577, 15905, 15759, 15617, 15808, 15772, 16125, 16122, 15921, 16307, 17196, 16866, 16854, 16783, 16896, 16874, 17688, 17427, 17382, 17649, 17690, 19749, 17752, 18119, 17821, 17749, 17711, 17959, 17851, 18016, 18493, 18187, 18742, 19193, 18970, 18859, 19120, 19451, 19255, 19329, 19561, 20247 1, 19965, 19828, 19744, 19876, 20188, 20041, 20429, 20340, 20284, 20343, 20602, 20519, 20460, 20549, 21082, 20934, 21041, 26290, 25967, 24223, 22637, 21524, 21238, 21107, 21331, 21279, 21482, 219 35, 21841, 21619, 21541, 21828, 21926, 22292, 22224, 22237, 22458, 22397, 22511, 23319, 23091, 23805, 22841, 22896, 23005, 23199, 23158, 23300, 23907, 23516, 23401, 23785, 23828, 23885, 24071, 2 4059, 24174, 24135, 24222, 27300, 25963, 25800, 24243, 24260, 24233, 24264, 24844, 24607, 24551, 24956, 25412, 25099, 25094, 25163, 25309, 25634, 25430, 25711, 26781, 26277, 26143, 26082, 25995, 26154, 26474, 26350, 26417, 26897, 26815, 26791, 26853, 27045, 26996, 26913, 27109, 27061, 28791, 27862, 27699, 27415, 27345, 27432, 27625, 27776, 27773, 27802, 28357, 28128, 28095, 28241, 2837 0, 28275, 28555, 28770, 29092, 29061, 28818, 29030, 29091, 29324, 29116, 29405, 29428, 31841, 30766, 30150, 29901, 29849, 29670, 29830, 29880, 30057, 30118, 30384, 30232, 30216, 30246, 30456, 30 423, 30479, 30613, 31518, 31035, 30856, 30968, 31306, 31693, 31612, 35217, 33728, 32688, 32457, 32012, 31886, 32311, 32018, 32575, 32571, 32637, 32803, 32790, 32756, 33237, 33121, 33683, 34440, 33737, 33730, 34394, 34906, 34801, 34863, 35154, 35195, 35648, 35408, 35273, 35263, 35520, 35478, 35646, 35981, 35784, 35671, 35859, 36242, 36072, 36009, 36105, 36281, 42298, 38981, 38247, 37563 3, 36673, 36494, 36621, 37061, 36837, 37194, 37284, 37829, 37613, 37604, 37567, 37804, 37811, 37894, 37856, 38021, 37920, 38154, 38545, 38305, 38266, 38429, 38461, 38388, 38763, 38666, 379 74, 38936, 39995, 39887, 39545, 39419, 39371, 39470, 39814, 39777, 39862, 39939, 39886, 39983, 41000, 40373, 40165, 40078, 40211, 40216, 40733, 40431, 40378, 40673, 40852, 40835, 40935, 42042, 4 1349, 41148, 41051, 41227, 41878, 41628, 41945, 42194, 42043, 42148, 42234, 45842, 41419, 43162, 43850, 42544, 42384, 42857, 42592, 42969, 43150, 43150, 43778, 43406, 43365, 43514, 43778, 43860, 43810, 43900, 44038, 44082, 44584, 44348, 44420, 44469, 44470, 44644, 44593, 44719, 44659, 44759, 45610, 45454, 45293, 45543, 45825, 45721, 46359, 46190, 46587, 45857, 45993, 45873, 4521 0, 46301, 46202, 47020, 46666, 46438, 46432, 46485, 46657, 46538, 46720, 46684, 46980, 46933, 47161, 47070, 47129, 47364, 47197, 47383, 76789, 65013, 61791, 51073, 48711, 48179, 47008, 47722, 47 439, 47853, 47967, 47915, 47847, 48425, 48262, 48194, 48518, 48445, 49573, 48452, 48805, 48768, 48770, 48949, 49383, 49359, 49269, 49512, 45401, 50208, 49860, 49715, 49835, 50099, 50000, 50665, 50305, 50209, 50851, 50667, 50809, 53817, 52629, 51778, 51316, 51283, 51207, 51545, 51407, 51748, 52200, 52040, 52626, 52411, 53226, 52861, 52814, 52700, 52815, 53119, 53695, 53623, 53120 5, 53330, 53323, 53238, 53888, 53348, 53754, 54126, 54026, 53881, 53818, 54052, 54078, 54483, 54415, 54346, 54451, 54772, 54589, 54538, 54906, 54918, 60275, 56784, 55860, 55551, 55366, 55250, 561 39, 55340, 55427, 55379, 55661, 55598, 55744, 55750, 56486, 56071, 56165, 56686, 56091, 56594, 56724, 58766, 57720, 57149, 56906, 56884, 56937, 57102, 57519, 57381, 57423, 57540, 57835, 57817, 5 7816, 58544, 58544, 58035, 58211, 58737, 58833, 59546, 59286, 58967, 58855, 59070, 59004, 59189, 59300, 59328, 59818, 59757, 59644, 60196, 59815, 60228, 62340, 60946, 60645, 60482, 60290, 60569, 60563, 60642, 60698, 60829, 61083, 61014, 61132, 60973, 61325, 61619, 61538, 61460, 61780, 62241, 62035, 62024, 62363, 63843, 63346, 62491, 61378, 62377, 62420, 62750, 62695, 62588, 62894, 6324 6, 63403, 63378, 63429, 63709, 64512, 64441, 63965, 63884, 64085, 64520, 64471, 64523, 64770, 64655, 64607, 64772, 64919, 64920, 71195, 68458, 66609, 65832, 65539, 65323, 65194, 65 042, 65259, 65516, 65506, 65601, 65596, 65550, 65778, 66195, 66133, 65923, 66502, 66447, 66621, 66456, 66586, 67609, 67240, 66949, 66735, 66638, 66776, 67092, 67530, 67419, 67540, 67938, 67635, 67628, 67936, 68312, 68078, 67972, 68188, 68317, 68669, 68801, 68802, 68562, 68520, 68668, 68872, 69651, 68941, 69283, 69608, 69501, 69538, 69653, 70729, 70390, 70602, 69761, 70121, 70396 3, 70717, 70914, 70815, 70769, 70825, 71108, 70976, 70960, 71192, 74283, 73226, 71841, 71675, 71494, 71368, 71442, 71556, 71665, 71747, 71706, 71777, 71784, 72549, 72265, 71874, 72527, 72286, 725 41, 72773, 72576, 72591, 72819, 73805, 73401, 73352, 73407, 73615, 73594, 73989, 73884, 73835, 73909, 74120, 74096, 74096, 74208, 75780, 75205, 74760, 74536, 74683, 7 4890, 74839, 74772, 74895, 74806, 75153, 75464, 75282, 75276, 75318, 75302, 75377, 75733, 75466, 75544, 76275, 75834, 76450, 76271, 76183, 75894, 76244, 76192, 76400, 76390, 76641, 76485, 76594, 7 76717, 76693, 76740, 80988, 83572, 80958, 78864, 77613, 77058, 76925, 76873, 77630, 77050, 77357, 77297, 77282, 77336, 77581, 77511, 78499, 77966, 77879, 77948, 78003, 78165, 78499, 78736, 78761, 78787, 78826, 78862, 78864, 7886 6, 78787, 78862, 79640, 79445, 79029, 78948, 78865, 79027, 79398, 79512, 79295, 79438, 79402, 79444, 79554, 79582, 79605, 79579, 79638, 80135, 79944, 79943, 79633, 79949, 79947, 80056, 80077, 80 42, 80269, 80607, 80764, 82255, 81497, 81132, 80988, 80981, 81034, 81124, 81273, 81202, 81272, 81318, 82045, 82016, 81761, 82026, 82033, 82078, 82062, 82170, 82418, 82317, 82417, 83172, 82954, 82950 3, 83080, 83080, 83148, 83302, 83320, 86487, 85832, 84315, 83913, 83880, 83800, 83859, 83823, 83911, 84128, 84825, 84008, 84172, 84102, 84758, 84504, 84361, 84495, 84691, 84179, 84966, 85025, 85515 8, 85466, 85364, 85216, 85522, 85521, 86150, 85913, 85657, 85607, 85790, 85924, 85918, 86157, 86342, 86404, 86373, 87644, 87227, 86946, 86617, 86492, 86625, 87218, 87166, 87044, 87218, 87464, 873 84, 87742, 87453, 87581, 87814, 88739, 88739, 88190, 88079, 87760, 87769, 88100, 88108, 88558, 88558, 88623, 88246, 88948, 88864, 88982, 88940, 89508, 89508, 89628, 89632, 8 9800, 89760, 89756, 89831, 89812, 89913, 95108, 93389, 91643, 91347, 90661, 90359, 90209, 90085, 90218, 90505, 90552, 91061, 90896, 90822, 91190, 91564, 91498, 91400, 91534, 91583, 91642, 92540, 92 92202, 92035, 91778, 91959, 92141, 92322, 92207, 92307, 92416, 92885, 92607, 92592, 92868, 93304, 94333, 93774, 93697, 93562, 93498, 93619, 93769, 93704, 94093, 94053, 94242, 94924, 9470 4, 94394, 94361, 94641, 94736, 95034, 94944, 95053, 97626, 96424, 95535, 95193, 95180, 95112, 95185, 95281, 95486, 96058, 95603, 95569, 96042, 95898, 96160, 96145, 96067, 96162, 96810, 96578, 96 441, 96759, 96711, 96793, 97308, 97230, 97160, 97248, 97285, 97587, 97587, 97626, 97732, 97822, 98241, 98242, 98393, 98584, 98644, 98698, 98709, 98937, 98987, 98992, 99125, 99246, 99534, 99345, 99300, 99268, 99479, 99456, 99532, 99758, 99694, 99546, 99706, 99740, 99901, 99759, 99798, 99997, 99925,

"C:\Users\cervat\Escritorio\F.Prog\Estructura de datos\arbol\AVL\bin\Debug\arbolAVL.exe"

99534, 99345, 99300, 99268, 99479, 99456, 99532, 99758, 99694, 99546, 99706, 99740, 99901, 99759, 99798, 99997, 99925,

RECORRIDO IN-ORDER

46, 76, 102, 157, 218, 351, 353, 387, 537, 538, 582, 654, 995, 999, 1001, 1174, 1376, 1420, 1545, 1546, 1577, 1661, 1662, 1670, 1698, 1729, 2045, 2113, 2171, 2199, 2226, 2614, 2686, 2 738, 2811, 2919, 3138, 3280, 3202, 3251, 3271, 3368, 3522, 3648, 3680, 3702, 3804, 4111, 4187, 4336, 4466, 4563, 4585, 4767, 4846, 4958, 4990, 5067, 5073, 5083, 5144, 5226, 5241, 5262, 5372, 540 0, 5538, 5656, 5789, 5908, 6042, 6055, 6128, 6198, 6225, 6348, 6383, 6394, 6652, 6779, 6873, 6876, 6924, 7052, 7337, 7561, 7689, 7794, 7842, 7974, 8056, 8097, 8121, 8156, 8181, 8195, 8404, 8646, 8 8701, 8751, 8765, 8871, 8927, 9248, 9258, 9280, 9458, 9600, 9628, 9648, 9836, 9760, 9884, 24408, 12883, 11709, 11005, 10595, 10323, 10041, 9950, 10107, 10339, 10543, 10675, 10796, 11278, 11281, 11288, 11296, 11321, 11335, 11411, 11509, 11793, 11867, 11872, 11922, 12003, 12034, 12124, 12146, 12167, 12439, 12627, 12835, 12840, 12883, 12924, 12947, 13038, 13054, 1307 0, 13171, 13523, 13591, 13674, 13753, 13777, 13787, 13790, 14022, 14066, 14107, 14116, 14192, 14229, 14261, 14293, 14314, 14346, 14408, 14427, 14450, 14551, 14551, 14673, 14676, 14689, 14689, 14 943, 15017, 15073, 15107, 15161, 15206, 15268, 15493, 15592, 15617, 15759, 15772, 15808, 15905, 15924, 16122, 16125, 16307, 16577, 16783, 16854, 16866, 16874, 16896, 17196, 17382, 17417, 17649, 17688, 17690, 17748, 17749, 17752, 17851, 17959, 18016, 18187, 18742, 18742, 18970, 18859, 18970, 19120, 19451, 19255, 19329, 19561, 20247 1, 19965, 20041, 20188, 20247, 20284, 20340, 20343, 20602, 20519, 20460, 20549, 21082, 20934, 21041, 26290, 25967, 24223, 22637, 21524, 21238, 21107, 21331, 21279, 21482, 219 35, 21841, 21619, 21541, 21828, 21926, 22292, 22224, 22237, 22458, 22397, 22511, 23319, 23091, 23805, 22841, 22896, 23005, 23199, 23158, 23300, 23907, 23516, 23401, 23785, 23828, 23885, 24071, 2 4059, 24174, 24135, 24222, 27300, 25963, 25800, 24243, 24260, 24233, 24264, 24844, 24607, 24551, 24956, 25412, 25099, 25094, 25163, 25309, 25634, 25430, 25711, 26781, 26277, 26143, 26082, 25995, 26154, 26474, 26350, 26417, 26897, 26815, 26791, 26853, 27045, 26996, 26913, 27109, 27061, 28791, 27862, 27699, 27415, 27345, 27432, 27625, 27776, 27773, 27802, 28357, 28128, 28095, 28241, 2837 0, 28275, 28555, 28770, 29092, 29061, 28818, 29030, 29091, 29324, 29116, 29405, 29428, 29567, 29670, 29830, 29880, 30057, 30118, 30150, 30216, 30232, 30246, 30384, 30423, 30456, 30479, 30613, 30766, 30 856, 30968, 31035, 31306, 31518, 31603, 31612, 31841, 31886, 32012, 32018, 32457, 32571, 32575, 32637, 32688, 32756, 32790, 32801, 33121, 33237, 33683, 33728, 33730, 33737, 34394, 34440, 34801, 34863, 34906, 35154, 35195, 35217, 35263, 35273, 35408, 35478, 35520, 35646, 35648, 35671, 35784, 35859, 35981, 36009, 36072, 36105, 36242, 36281, 36290, 36294, 36621, 36673, 37061 3, 37184, 37194, 37563, 37567, 37604, 37604, 37613, 37811, 37829, 37858, 37894, 37920, 38021, 38154, 38247, 38266, 38305, 38429, 38454, 38461, 38545, 38666, 38763, 38838, 38936, 38974, 38981, 393 72, 39419, 39470, 39545, 39777, 39814, 39862, 39867, 39886, 39939, 39983, 40078, 40165, 40211, 40216, 40733, 40431, 40378, 40673, 40852, 40835, 40935, 41000, 41051, 41148, 41227, 4 1349, 41628, 41878, 41945, 42042, 42043, 42169, 42194, 42234, 42298, 42304, 42514, 42592, 42597, 42969, 43080, 43150, 43150, 43162, 43365, 43406, 43514, 43778, 43810, 43860, 43878, 43900, 44038, 44193, 44220, 44348, 44469, 44470, 44584, 44593, 44644, 44659, 44719, 44759, 44844, 44959, 45293, 45454, 45610, 45721, 45825, 45842, 45857, 45873, 45993, 46120, 46190, 46202, 4630 1, 46359, 46405, 46432, 46438, 46538, 46567, 46666, 46684, 46720, 46933, 46980, 47020, 47070, 47129, 47161, 47197, 47364, 47



Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
ICOM – Ingeniería en Computación
Módulo Estructura de Datos



"C:\Users\cervat\Escritorio\F.Prog.Estructura de datos\arbol\AW\bin\Debug\arbol\AW.exe"

RECORRIDO POST-ORDER

```
76, 46, 120, 157, 182, 353, 351, 537, 582, 538, 387, 218, 999, 995, 1174, 1376, 1881, 1545, 1577, 1546, 1662, 1670, 1661, 1420, 2845, 1729, 2171, 2614, 2226, 2199, 2113, 1698, 654, 2696, 2811, 3
150, 2919, 2730, 3202, 3251, 3368, 3522, 3271, 3200, 3680, 4004, 3732, 4187, 4336, 4111, 4563, 4563, 4046, 4767, 4466, 3648, 2680, 4908, 5073, 5083, 5067, 5226, 5241, 5372, 5538, 5408, 5262, 514
4, 5789, 5908, 6055, 6128, 6042, 6225, 6383, 6348, 6198, 5656, 6652, 6779, 6876, 6873, 7852, 7689, 7561, 7337, 7842, 7842, 8056, 7974, 8121, 8156, 8195, 8161, 8097, 7794, 8046, 8161, 8765, 8871,
9240, 9258, 8927, 8751, 9600, 9459, 9760, 9894, 9836, 9628, 9280, 8404, 6394, 4958, 9950, 10107, 10041, 18543, 10339, 10323, 10796, 10675, 10595, 11014, 11102, 11066, 11259, 11249, 11281, 11296
, 11288, 11335, 11509, 11421, 11321, 11278, 11095, 11793, 11922, 11872, 11867, 12094, 12124, 12093, 12167, 12627, 12840, 12835, 12439, 12146, 11799, 12524, 12947, 13054, 13098, 13171, 13523, 130
70, 13753, 13674, 13787, 13777, 14022, 14107, 14066, 13790, 14198, 14261, 14229, 14314, 14346, 14293, 14116, 13591, 12883, 14427, 14450, 14551, 14676, 14673, 14858, 15017, 14943, 14680, 14551, 1
5187, 15161, 15260, 15493, 15206, 15073, 15817, 15772, 15808, 15759, 15921, 16122, 16307, 16125, 15905, 16783, 16854, 16874, 16896, 16866, 17382, 17649, 17417, 17690, 17688, 17196, 16577, 15592,
17771, 17749, 17851, 18016, 17959, 17821, 18415, 18187, 18742, 18737, 18493, 18119, 18859, 19120, 18970, 19329, 19255, 19561, 19451, 19193, 18752, 19744, 19876, 19828, 20041, 20188, 19965, 2028
4, 20343, 20340, 20460, 20549, 20519, 20934, 21041, 21002, 20602, 20429, 20247, 19704, 17709, 14408, 9901, 21107, 21279, 21482, 21331, 21230, 21541, 21828, 21619, 21926, 21841, 22237, 22224, 223
97, 22551, 22458, 22292, 21935, 21524, 22896, 22841, 23005, 23005, 23158, 23300, 23199, 23091, 23401, 23528, 23885, 23785, 23516, 24059, 24135, 24222, 24174, 24071, 23907, 23319, 22637, 24233, 2
4264, 24260, 24551, 24607, 24956, 24844, 24352, 25094, 25309, 25163, 25099, 25430, 25711, 25634, 25412, 25800, 25995, 26002, 26154, 26143, 26350, 26417, 26404, 26277, 26791, 26853, 26815, 26913,
27019, 26996, 27061, 27045, 26897, 26781, 25963, 27345, 27625, 27432, 27415, 27773, 27802, 27776, 27699, 28095, 28370, 28241, 28120, 28555, 28770, 28729, 28397, 27852, 29030, 28818, 29091, 2906
1, 29516, 29428, 29405, 29324, 29092, 28791, 27300, 24223, 29830, 29670, 29880, 29849, 30118, 30057, 30216, 30246, 30232, 30423, 30613, 30479, 30456, 30384, 30150, 30968, 30856, 31306, 31
035, 31612, 31603, 31518, 30766, 31886, 32018, 32311, 32012, 32571, 32637, 32575, 32457, 32756, 32790, 33121, 33683, 33237, 32801, 32688, 33738, 34394, 33737, 34863, 34881, 35195, 35154, 34906,
34440, 33728, 35263, 35273, 35478, 35646, 35520, 35408, 35671, 35859, 35784, 36009, 36105, 36072, 36281, 36242, 35981, 35648, 35217, 31841, 29567, 36621, 36494, 36837, 37184, 37194, 37061, 36673
, 37567, 37604, 37604, 37811, 37613, 37856, 37928, 38154, 38021, 37894, 37829, 37565, 38266, 38429, 38305, 38461, 38454, 38666, 38763, 38936, 38974, 38838, 38545, 38247, 39372, 39470, 39419, 397
77, 39862, 39814, 39545, 39885, 39933, 39867, 40078, 40216, 40211, 40165, 40378, 40673, 40431, 40835, 40935, 40852, 40733, 40373, 41051, 41277, 41148, 41628, 41945, 41878, 41349, 42169, 4
2043, 42234, 42104, 42042, 41080, 39995, 38081, 42304, 42582, 42989, 42857, 42514, 43150, 43150, 43009, 43365, 43514, 43406, 43810, 43860, 44038, 43900, 43678, 43778, 43362, 44220, 44470, 44469,
44340, 44593, 44659, 44759, 44719, 44644, 44584, 45275, 45293, 45543, 45445, 45721, 45825, 45610, 44892, 44193, 45857, 45873, 46120, 45993, 45867, 46202, 46301, 46190, 46405, 46432, 46538, 4665
7, 46438, 46684, 46933, 46980, 46720, 46666, 47129, 47070, 47197, 47383, 47364, 47161, 47020, 46359, 45842, 42298, 36290, 21876, 47439, 47853, 47772, 47943, 47915, 48097, 47987, 47908, 48194, 48
262, 48445, 48518, 48425, 48179, 48770, 48768, 48949, 48805, 49269, 49359, 49401, 49512, 49583, 49052, 49835, 49715, 50090, 50029, 49860, 50209, 50305, 50667, 50899, 50851, 50665, 50208, 49973, 5
48711, 51107, 51283, 51407, 51748, 51545, 51316, 52040, 52411, 52626, 52209, 51778, 52708, 52815, 52814, 53023, 53099, 53120, 53119, 52861, 53238, 53323, 53348, 53754, 53588, 53336, 53226, 52629
, 53818, 53881, 54078, 54852, 54025, 54346, 54451, 54415, 54538, 54589, 54918, 54906, 54772, 54483, 54126, 53817, 51073, 55239, 55340, 55250, 55379, 55427, 55366, 55598, 55750, 55744, 55661, 555
51, 56165, 56071, 56504, 56491, 56724, 56686, 56486, 56937, 56884, 57102, 56986, 57423, 57381, 57540, 57519, 57149, 57816, 57817, 58035, 58231, 58214, 58633, 58737, 58544, 57835, 57720, 5
8855, 59004, 59189, 59079, 58967, 59328, 59300, 59286, 59644, 59757, 59825, 60220, 60196, 59818, 59546, 58766, 56784, 60290, 60563, 60642, 60569, 60482, 60829, 60698, 60645, 60973, 61325, 61312,
61460, 61538, 61780, 61619, 61414, 62024, 62035, 62303, 62241, 61803, 60946, 62377, 62420, 62378, 62508, 62695, 63246, 62894, 62750, 62491, 63371, 63378, 63429, 63709, 63642, 63403, 63346, 6388
4, 64005, 63965, 64471, 64523, 64520, 64441, 64687, 64772, 64655, 64920, 64919, 64797, 64542, 63843, 62340, 60275, 54971, 65042, 65259, 65194, 65506, 65516, 65323, 65550, 65596, 65778, 65601, 65
539, 65923, 66131, 66251, 66456, 66447, 66586, 66502, 66195, 65832, 66638, 66776, 66735, 67092, 66949, 67419, 67540, 67530, 67224, 67628, 67936, 67635, 67972, 68188, 68070, 68317, 68312, 67938,
67609, 66609, 68520, 68669, 68562, 68872, 68868, 68802, 68903, 69041, 69538, 69501, 69653, 69608, 69368, 68901, 69761, 70121, 70092, 70717, 70396, 70390, 70769, 70825, 70815, 70906, 70976, 71192
71108, 70914, 70729, 69669, 68458, 71442, 71346, 71605, 71556, 71469, 71706, 71778, 71777, 71747, 71675, 71874, 72296, 72541, 72527, 72265, 72591, 72576, 72831, 72941, 72850, 72773, 72549, 718
41, 73352, 73437, 73461, 73594, 73615, 73519, 73835, 73909, 73884, 74096, 74096, 74288, 74128, 73899, 73805, 73226, 74476, 74683, 74536, 74772, 74839, 74896, 75115, 74995, 74890, 74766, 75266, 7
5276, 75302, 75377, 75318, 75282, 75544, 75625, 75763, 75729, 75464, 75205, 75894, 76192, 76244, 76183, 76398, 76400, 76271, 76584, 76485, 76693, 76740, 76717, 76641, 76450, 75834, 74283, 71195, 7
6013, 76873, 77050, 77030, 76925, 77282, 77336, 77297, 77511, 77561, 77557, 77058, 77548, 77879, 78165, 78003, 77966, 78736, 78787, 78761, 78862, 78826, 78499, 77613, 78865, 79027, 78948, 7929
5, 79352, 79402, 79444, 79436, 79358, 79029, 79502, 79579, 79630, 79605, 79554, 79445, 79683, 79943, 79947, 80077, 80055, 79949, 79944, 80269, 80764, 80807, 80422, 80135, 79540, 78864, 80981, 81
124, 81634, 80908, 81272, 81202, 81318, 81273, 81132, 81761, 82033, 82026, 82016, 82062, 82170, 82070, 82045, 81497, 82417, 82317, 82950, 83080, 82954, 83302, 83520, 83461, 83172, 82418, 82255,
80956, 83659, 83823, 83800, 83911, 83880, 84008, 84025, 84202, 84172, 84128, 83913, 84495, 84361, 84691, 84504, 84966, 85025, 84977, 84758, 84315, 85216, 85364, 85521, 85522, 85466, 85607, 85790
, 85657, 85919, 85924, 85913, 86342, 86373, 86404, 86357, 86158, 85554, 85892, 86492, 86625, 86617, 87044, 87166, 87218, 87218, 86946, 87244, 87453, 87384, 87491, 87636, 87581, 87464, 87227, 877
09, 87760, 88158, 88100, 88079, 88263, 88655, 88733, 88667, 88558, 88190, 88864, 89043, 88982, 88948, 89423, 89632, 89628, 89508, 89756, 89760, 89812, 89913, 89831, 89800, 89650, 89246, 88739, 8
7644, 86487, 83572, 90085, 90218, 90209, 90552, 90505, 90359, 90822, 90896, 91190, 91061, 90661, 91400, 91534, 91498, 91642, 91583, 91564, 91347, 91959, 91778, 92141, 92035, 92207, 92416, 92307, 9
2232, 92282, 92592, 92868, 92607, 93204, 92885, 92540, 91643, 93498, 93619, 93562, 93704, 93769, 93697, 93875, 94053, 94242, 94093, 93774, 94361, 94641, 94394, 94736, 94704, 94944, 95053, 9503
4, 94924, 94333, 93389, 95112, 95185, 95180, 95486, 95281, 95193, 95568, 95898, 96042, 95603, 96067, 96145, 96162, 96160, 96058, 95535, 96441, 96711, 96793, 96759, 96578, 97160, 97230, 97285, 97
274, 97501, 97587, 97435, 97258, 96810, 96424, 97732, 98241, 98242, 97822, 98584, 98644, 98709, 98937, 98658, 98992, 99060, 99125, 99098, 98987, 98393, 99268, 99300, 99456, 99532, 99479, 99345,
99546, 99740, 99706, 99694, 99798, 99759, 99925, 99997, 99901, 99758, 99534, 99246, 97626, 95108, 89988, 76789, 47408,
```

"C:\Users\cervat\Escritorio\F.Prog.Estructura de datos\arbol\AW\bin\Debug\arbol\AW.exe"

```
5276, 76302, 75377, 75318, 75282, 75544, 75625, 75763, 75729, 75464, 75205, 75894, 76192, 76244, 76183, 76398, 76400, 76271, 76584, 76485, 76693, 76740, 76717, 76641, 76450, 75834, 74283, 71195, 7
6013, 76873, 77050, 77030, 76925, 77282, 77336, 77297, 77511, 77561, 77557, 77058, 77548, 77879, 78165, 78003, 77966, 78736, 78787, 78761, 78862, 78826, 78499, 77613, 78865, 79027, 78948, 7929
5, 79352, 79402, 79444, 79436, 79358, 79029, 79502, 79579, 79630, 79605, 79554, 79445, 79683, 79943, 79947, 80077, 80055, 79949, 79944, 80269, 80764, 80807, 80422, 80135, 79540, 78864, 80981, 81
124, 81634, 80908, 81272, 81202, 81318, 81273, 81132, 81761, 82033, 82026, 82016, 82062, 82170, 82070, 82045, 81497, 82417, 82317, 82950, 83080, 82954, 83302, 83520, 83461, 83172, 82418, 82255,
80956, 83659, 83823, 83800, 83911, 83880, 84008, 84025, 84202, 84172, 84128, 83913, 84495, 84361, 84691, 84504, 84966, 85025, 84977, 84758, 84315, 85216, 85364, 85521, 85522, 85466, 85607, 85790
, 85657, 85919, 85924, 85913, 86342, 86373, 86404, 86357, 86158, 85554, 85892, 86492, 86625, 86617, 87044, 87166, 87218, 87218, 86946, 87244, 87453, 87384, 87491, 87636, 87581, 87464, 87227, 877
09, 87760, 88158, 88100, 88079, 88263, 88655, 88733, 88667, 88558, 88190, 88864, 89043, 88982, 88948, 89423, 89632, 89628, 89508, 89756, 89760, 89812, 89913, 89831, 89800, 89650, 89246, 88739, 8
7644, 86487, 83572, 90085, 90218, 90209, 90552, 90505, 90359, 90822, 90896, 91190, 91061, 90661, 91400, 91534, 91498, 91642, 91583, 91564, 91347, 91959, 91778, 92141, 92035, 92207, 92416, 92307, 9
2232, 92282, 92592, 92868, 92607, 93204, 92885, 92540, 91643, 93498, 93619, 93562, 93704, 93769, 93697, 93875, 94053, 94242, 94093, 93774, 94361, 94641, 94394, 94736, 94704, 94944, 95053, 9503
4, 94924, 94333, 93389, 95112, 95185, 95180, 95486, 95281, 95193, 95568, 95898, 96042, 95603, 96067, 96145, 96162, 96160, 96058, 95535, 96441, 96711, 96793, 96759, 96578, 97160, 97230, 97285, 97
274, 97501, 97587, 97435, 97258, 96810, 96424, 97732, 98241, 98242, 97822, 98584, 98644, 98709, 98937, 98658, 98992, 99060, 99125, 99098, 98987, 98393, 99268, 99300, 99456, 99532, 99479, 99345,
99546, 99740, 99706, 99694, 99798, 99759, 99925, 99997, 99901, 99758, 99534, 99246, 97626, 95108, 89988, 76789, 47408,
```

----- ELEMENTO MENOR EN EL ARBOL -----

- 46

----- ELEMENTO MAYOR EN EL ARBOL -----

- 99997

----- ALTURA DEL ARBOL -----

- 12

----- ALTURA DEL SUB-ARBOL IZQUIERDO -----

- 11

----- ALTURA DEL SUB-ARBOL DERECHO -----

- 11

Eliminando arbol...

Se elimino correctamente :)

Press returned 0 (0x0) execution time : 5.738 s
Press any key to continue.



Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
ICOM – Ingeniería en Computación
Módulo Estructura de Datos



Prueba 3 con AVL:

```
"C:\Users\cerva\Escritorio\F.Prog\Estructura de datos\arbol\AVL\bin\Debug\arbol\AVL.exe"
Cuantos numeros desea insertar en el arbol?
1888

Insertando: 66184
Insertando: 81335
Insertando: 46958
Insertando: 93896
Insertando: 51991
Insertando: 83184
Insertando: 90834
Insertando: 76889
Insertando: 2535
Insertando: 7909
Insertando: 33630
Insertando: 1375
Insertando: 14951
Insertando: 76919
Insertando: 28747
Insertando: 34458
Insertando: 19064
Insertando: 97527
Insertando: 86176
Insertando: 14762
Insertando: 95398
Insertando: 94724
Insertando: 72557
Insertando: 32120
Insertando: 36266
Insertando: 8565
Insertando: 61988
Insertando: 94853
Insertando: 6965
Insertando: 65888
Insertando: 13743
Insertando: 76648
Insertando: 9725
Insertando: 46163
Insertando: 39868
Insertando: 39303
Insertando: 49195
Insertando: 201
Insertando: 91891
Insertando: 64542
Insertando: 28479
Insertando: 35578
Insertando: 18387
Insertando: 25369
Insertando: 74426
Insertando: 41934
Insertando: 31798
Insertando: 3965
Insertando: 47419
```

```
"C:\Users\cerva\Escritorio\F.Prog\Estructura de datos\arbol\AVL\bin\Debug\arbol\AVL.exe"
----- RECORRIDO PRE-ORDER -----
36266, 19064, 7909, 3965, 2535, 1375, 396, 235, 261, 372, 314, 807, 474, 463, 758, 1002, 2099, 1935, 1861, 1452, 1899, 1888, 1953, 2231, 2209, 2158, 2362, 3518, 3154, 2678, 2563, 2924, 2786, 310
5, 3222, 3449, 3957, 3627, 3539, 3657, 3959, 6688, 5387, 4521, 4294, 4095, 4464, 4749, 4573, 4553, 4841, 4937, 5770, 5613, 5487, 5421, 5581, 5680, 5616, 5735, 6510, 6059, 5926, 6312, 6127, 6577,
6566, 6593, 7246, 6965, 6783, 6705, 6746, 6903, 7071, 7025, 7128, 7805, 7372, 7273, 7383, 7832, 7856, 12454, 9567, 9028, 8565, 8261, 8085, 8237, 8484, 8362, 8521, 8888, 8693, 8747, 8977, 9406,
9184, 9126, 9267, 9503, 9551, 10256, 10108, 9856, 9716, 9676, 9725, 9860, 10165, 10126, 10198, 11280, 11055, 10642, 10661, 11203, 11098, 11233, 11275, 11571, 11381, 11411, 12051, 11655, 12154, 1
7116, 14762, 14064, 13214, 12826, 12788, 12934, 12894, 12947, 13483, 13381, 13327, 13647, 13547, 13743, 14310, 14155, 14103, 14229, 14157, 14695, 14405, 14653, 14752, 15474, 14951, 14861, 14849,
15255, 14963, 15140, 15436, 15293, 16008, 15640, 15592, 15645, 16633, 16594, 16355, 16744, 16684, 18064, 17659, 17362, 17196, 17571, 17957, 17814, 17712, 17815, 18042, 18715, 18346, 18388, 1838
7, 18748, 18975, 28747, 22639, 21758, 20146, 19493, 19141, 19118, 19445, 19989, 19851, 20047, 20949, 20551, 20445, 20263, 20646, 20687, 20671, 21408, 21211, 21107, 21649, 21470, 21727, 22494, 21
989, 21817, 22128, 22041, 22215, 22535, 22568, 22589, 25959, 25458, 23146, 23146, 23220, 22906, 31611, 23350, 23238, 23248, 23365, 23339, 23427, 24165, 24068, 23867, 24115, 24849, 24296, 24268,
24508, 25116, 25885, 25238, 26431, 25616, 25460, 25526, 26234, 26820, 25711, 26202, 26407, 27852, 27478, 26874, 26525, 26484, 26811, 27388, 27614, 27596, 27503, 27634, 27753, 28409, 28148, 27911
, 28341, 28248, 28353, 28736, 28592, 28479, 28719, 28744, 33630, 31708, 30271, 29602, 29267, 29044, 29856, 29598, 29735, 29645, 30025, 29934, 31369, 31284, 30654, 30312, 31348, 31357, 31597, 314
11, 31654, 31620, 32335, 32128, 31941, 31745, 31730, 31791, 32049, 32046, 32074, 32471, 32287, 32130, 32382, 32573, 33125, 33045, 33063, 33261, 33233, 33198, 33248, 33575, 33543, 33606, 34629, 3
4081, 33990, 33686, 33830, 34028, 34024, 34222, 34107, 34458, 34467, 35570, 35433, 35003, 34723, 35417, 35494, 35543, 36006, 35693, 35574, 35873, 35972, 36180, 36048, 36245, 66104, 46958, 41895,
38556, 37999, 37359, 37051, 36699, 36468, 37129, 37113, 37515, 37391, 37757, 37771, 38370, 38236, 38167, 38258, 38465, 38419, 38493, 39068, 39303, 38945, 38635, 38578, 38756, 39096, 39046, 3919
4, 39580, 39552, 39703, 39800, 40747, 40419, 40065, 40032, 39978, 40202, 40097, 40333, 40504, 40424, 40629, 40553, 41221, 41094, 40998, 41091, 41373, 41267, 41484, 41594, 43738, 42843, 42603, 42
167, 42828, 41908, 42113, 42168, 42423, 42742, 42635, 42755, 43085, 42962, 43039, 43389, 43218, 43462, 43512, 44302, 43932, 43775, 43768, 43851, 44143, 44123, 44117, 44129, 44253, 44278, 45899,
45298, 44760, 44468, 44664, 45067, 45389, 45335, 45333, 45643, 46163, 46029, 45939, 46053, 46050, 46273, 46271, 46381, 46375, 46563, 56150, 51991, 49195, 48046, 47419, 47097, 46967, 47023, 47328
, 47126, 47773, 47745, 47944, 48931, 48587, 48608, 49059, 48952, 50821, 49854, 49598, 49495, 49225, 49653, 50445, 49998, 49669, 50630, 50640, 51478, 50998, 50988, 50883, 51343, 51052, 51610, 515
75, 51888, 51827, 54425, 93061, 52342, 52217, 52141, 52133, 52177, 52291, 52754, 52443, 52367, 52907, 53762, 53213, 53067, 53181, 53615, 53579, 54054, 53854, 54148, 54240, 55152, 54578, 54522, 5
4522, 54766, 54687, 55084, 54829, 55095, 55530, 55477, 55393, 55587, 56084, 55962, 55810, 56182, 60976, 58087, 56930, 56588, 56474, 56249, 56210, 56395, 56567, 56590, 56835, 57240, 57012, 56983,
57185, 57167, 57629, 57281, 57672, 58058, 59742, 58787, 58588, 58327, 58387, 58328, 58630, 58726, 58846, 58828, 59002, 58939, 59327, 60200, 59885, 59797, 60096, 60048, 60157, 60515, 60417, 6051
1, 60760, 60645, 60815, 60791, 63561, 61988, 61477, 61369, 61140, 61091, 61192, 61398, 61414, 61596, 61576, 61684, 62904, 62722, 62468, 62015, 62765, 62820, 63436, 63309, 63165, 63333, 63474, 64
474, 63991, 63895, 63773, 63941, 64385, 64294, 64078, 64324, 64406, 65257, 64715, 64542, 64665, 65110, 65008, 64945, 65247, 65747, 65562, 65358, 65739, 65804, 65763, 65800, 65865, 83104, 74426,
69832, 67632, 68787, 68445, 68426, 68140, 68666, 67951, 68844, 68840, 67378, 67219, 67526, 68348, 67783, 67675, 67655, 67747, 68277, 67971, 68322, 68612, 69421, 68530, 69537, 68726, 68643, 69828
, 71905, 71899, 70346, 69998, 69999, 69937, 70232, 70115, 70267, 70633, 70530, 70542, 70682, 71005, 71531, 71212, 71072, 71206, 71521, 71313, 71856, 71729, 71565, 71891, 73161, 72879, 72544, 722
57, 71953, 72359, 72557, 72782, 73853, 72984, 72976, 73035, 73097, 73122, 73963, 73631, 73254, 73242, 73768, 73752, 74104, 74065, 73987, 74293, 74491, 75211, 74857, 74672, 74456, 74785, 74781, 7
4847, 74980, 74927, 75096, 76668, 75627, 75448, 75448, 75439, 75527, 75489, 75612, 75693, 75640, 76222, 75720, 76371, 77073, 76918, 76899, 76718, 76971, 76932, 77442, 77339, 77095, 77488, 80253,
78749, 78120, 77781, 77536, 77727, 78083, 78593, 78370, 78144, 78741, 78743, 79632, 79087, 79009, 79803, 79458, 79943, 79882, 80120, 81533, 80965, 80541, 80480, 80435, 80569, 80662, 8086
6, 81335, 80968, 81471, 81423, 81485, 82531, 82053, 81679, 81618, 81943, 81721, 82314, 82120, 82495, 82617, 82575, 82571, 82551, 82576, 82698, 82621, 82655, 83081, 83006, 90934, 86500, 85
103, 84392, 84286, 83890, 83477, 83196, 83637, 83944, 83931, 84008, 84262, 84227, 84361, 84305, 84718, 84559, 84547, 84580, 84895, 84812, 84810, 84929, 84909, 85489, 85125, 85110, 85327, 85145,
85485, 86139, 85922, 85919, 85586, 86030, 86176, 86204, 88919, 87617, 87058, 86776, 86579, 86634, 87031, 86833, 87055, 87251, 87197, 87098, 87435, 87268, 87411, 87458, 87459, 88474, 88360, 87859
, 87855, 87825, 88175, 88329, 88441, 88409, 88462, 88550, 88541, 88859, 88599, 88862, 90353, 89767, 89269, 88937, 89011, 89761, 89484, 90086, 89775, 90036, 90300, 90086, 90348, 90809, 90502, 904
63, 90396, 90740, 90526, 90893, 90845, 90932, 94724, 92732, 91891, 91607, 91417, 91126, 91067, 91128, 91552, 91548, 91570, 91582, 91796, 91654, 91625, 91869, 92440, 92280, 92062, 92605, 92456, 9
2630, 92621, 92649, 93896, 93278, 93062, 92936, 92876, 93230, 93641, 93541, 93329, 93860, 93825, 94244, 94097, 93987, 93971, 94083, 94207, 94133, 94444, 94304, 94299, 94419, 94506, 97527, 96741,
95586, 95398, 94863, 94786, 94980, 94931, 95189, 95481, 95402, 95511, 95933, 95615, 95601, 95630, 95870, 96237, 96041, 96188, 96507, 96283, 96739, 97297, 97092, 96935, 97043, 97129, 97109, 9720
4, 97446, 97378, 97459, 98964, 98610, 97943, 97654, 97532, 97640, 97804, 98248, 98088, 97952, 98219, 98386, 98528, 98699, 98698, 98656, 98925, 98921, 98712, 98963, 99250, 99105, 99054, 98967, 99
199, 99146, 99247, 99441, 99283, 99266, 99254, 99377, 99323, 99390, 99741, 99732, 99889, 99959,
```




```
"C:\Users\cerva\Escritorio\F.Prog\Estructura de datos\arbol\AVL\bin\Debug\arbolAVL.exe"
283, 99732, 99959, 99869, 99741, 99441, 99250, 98964, 97527, 94724, 90934, 83104, 66104, 36266,

----- ELEMENTO MENOR EN EL ARBOL -----
- 281
----- ELEMENTO MAYOR EN EL ARBOL -----
- 99959
----- ALTURA DEL ARBOL -----
- 12
----- ALTURA DEL SUB-ARBOL IZQUIERDO -----
- 10
----- ALTURA DEL SUB-ARBOL DERECHO -----
- 11
Eliminando arbol...
Se eliminó correctamente :)
Process returned 0 (0x0)   execution time : 4.066 s
Press any key to continue.
```

Conclusión

Fue una práctica rápida de codificar, como ya tenía la base del trabajo anterior solo fue cuestión de agregar los métodos vistos en clase, guiándome de las diapositivas del profe; una vez terminado solo fue cuestión de agregar temporalmente una línea de código para imprimir y verificar el correcto funcionamiento del árbol con cada inserción, una vez depurado y hacer la verificación a mano, comprobando el correcto funcionamiento; las elimine y reduje algunas líneas de código para hacerlo más óptimo.

Me sirvió la practica para reforzar el tema del miércoles, al principio me confundía un poco, ya que tenia el orden corrector pero los enunciados los decía de manera diferente a las diapositivas; pero al final logre reforzar esta parte.