**Arboles**

En ciencias de la computación y en informática, un **árbol** es un tipo abstracto de datos (TAD) ampliamente usado que imita la estructura jerárquica de un árbol, con un valor en la raíz y subárboles con un nodo padre, representado como un conjunto de nodos enlazados.

**Preorden**: (**raíz**, izquierdo, derecho).

**Inorden**: (izquierdo, **raíz**, derecho)

**Postorden**: (izquierdo, derecho, **raíz**).

Definición de Nodo

struct Nodo {

int dato;

struct Nodo \*izquierda;

struct Nodo \*derecha;

};

Crear Nodo

struct Nodo \*nuevoNodo(int dato) {

// Solicitar memoria

size\_t tamanioNodo = sizeof(struct Nodo);

struct Nodo \*nodo = (struct Nodo \*) malloc(tamanioNodo);

// Asignar el dato e iniciar hojas

nodo->dato = dato;

nodo->izquierda = nodo->derecha = NULL;

return nodo;

}

Insertar

void insertar(struct Nodo \*nodo, int dato) {

// ¿Izquierda o derecha?

// Si es mayor va a la derecha

if (dato > nodo->dato) {

// Tienes espacio a la derecha?

if (nodo->derecha == NULL) {

nodo->derecha = nuevoNodo(dato);

} else {

// Si la derecha ya está ocupada, recursividad ;)

insertar(nodo->derecha, dato);

}

} else {

// Si no, a la izquierda

if (nodo->izquierda == NULL) {

nodo->izquierda = nuevoNodo(dato);

} else {

// Si la izquierda ya está ocupada, recursividad ;)

insertar(nodo->izquierda, dato);

}

}

}

Preorden

void preorden(struct Nodo \*nodo) {

if (nodo != NULL) {

printf("%d,", nodo->dato);

preorden(nodo->izquierda);

preorden(nodo->derecha);

}

}

Inorden

void inorden(struct Nodo \*nodo) {

if (nodo != NULL) {

inorden(nodo->izquierda);

printf("%d,", nodo->dato);

inorden(nodo->derecha);

}

}

Postorden

void postorden(struct Nodo \*nodo) {

if (nodo != NULL) {

postorden(nodo->izquierda);

postorden(nodo->derecha);

printf("%d,", nodo->dato);

}

}

Uso

int main(void) {

struct Nodo \*raiz = nuevoNodo(28);

insertar(raiz, 11);

insertar(raiz, 96);

insertar(raiz, 21);

insertar(raiz, 6);

insertar(raiz, 97);

insertar(raiz, 1);

insertar(raiz, 30);

insertar(raiz, 10);

insertar(raiz, 2);

printf("\nImprimiendo preorden\n");

preorden(raiz);

printf("\nImprimiendo inorden\n");

inorden(raiz);

printf("\nImprimiendo postorden\n");

postorden(raiz);

}