**UNIVERSIDAD ABIERTA PARA ADULTOS**

**Asignatura:**

Estructura de Datos y Algoritmos

**Tema:**

Seminario III

**Participantes:**

Christian Medrano

Johan García Ramírez

Ryan Rene Sánchez

Dawlin Diaz Figueroa

Rigoberto Rodriguez

Carlos Cueva

Francis Gómez

**Matriculas:**

1000-29253

1000-16023

1000-31794

1000-20389

1000-16189

1000-38032

1000-24212

**Facilitador:**

Maikel Aparicio

**Fecha:** 20/3/2022

* **Árboles binarios de búsqueda:**

Un árbol binario de búsqueda (ABB) es un árbol binario con la propiedad de que todos los elementos almacenados en el subárbol izquierdo de cualquier nodo x son menores que el elemento almacenado en x, y todos los elementos almacenados en el subárbol derecho de x son mayores que el elemento almacenado en x.

La búsqueda en árboles binarios es un método de búsqueda simple, dinámico y eficiente considerado como uno de los fundamentales en Ciencia de la Computación. De toda la terminología sobre árboles, tan sólo recordar que la propiedad que define un árbol binario es que cada nodo tiene a lo más un hijo a la izquierda y uno a la derecha. Para construir los algoritmos consideraremos que cada nodo contiene un registro con un valor clave a través del cual efectuaremos las búsquedas. En las implementaciones que presentaremos sólo se considerará en cada nodo del árbol un valor del tipo elemento, aunque en un caso general ese tipo estará compuesto por dos: una clave indicando el campo por el cual se realiza la ordenación y una información asociada a dicha clave o visto de otra forma, una información que puede ser compuesta en la cual existe definido un orden.

Es conveniente hacer notar la diferencia entre este procedimiento y el de búsqueda binaria. En éste podría pensarse en que se usa un árbol binario para describir la secuencia de comparaciones hecha por una función de búsqueda sobre el vector. En cambio, en los ABB se construye una estructura de datos con registros conectados por punteros y se usa esta estructura para la búsqueda. El procedimiento de construcción de un ABB puede basarse en un procedimiento de inserción que vaya añadiendo elementos al árbol. Tal procedimiento comenzaría mirando si el árbol es vacío y de ser así se crearía un nuevo nodo para el elemento insertado devolviendo como árbol resultado un puntero a ese nodo.

**Cont.**

Si el árbol no está vacío se busca el elemento a insertar como lo hace el procedimiento pertenece sólo que al encontrar un puntero NULL durante la búsqueda, se reemplaza por un puntero a un nodo nuevo que contenga el elemento a insertar.

**El código podría ser el siguiente:**

* **Ejemplo:**

void Inserta(tElemento x,ABB \*t)

{

if(!(\*t)){

\*t=(nodoABB)malloc(sizeof(struct tipoceldaABB));

if(!(\*t)){

error("Memoria Insuficiente.");

}

(\*t)->etiqueta=x;

(\*t)->hizqda=NULL;

(\*t)->hdrcha=NULL;

} else if(x<(\*t)->etiqueta)

inserta(x,&((\*t)->hizqda));

else

inserta(x,&((\*t)->hdrcha));