



ALGORITMOS Y ESTRUCTURAS DE DATOS II

Ingeniería Técnica en Informática de Gestión
Ingeniería Técnica en Informática de Sistemas

CURSO 2002/03

Implementación del TAD árbol binario de búsqueda (ABB)

módulo árbolesBinarios

importa defTipoElemento

exporta

tipo arbin

...

acción insertar (**var** a: arbin; e: elemento)

función está (a: arbin; e: elemento): booleano

función min (a: arbin): elemento

función max (a: arbin): elemento

acción borrar (**var** a: arbin; e: elemento)

implementación

tipos

arbin = **puntero a** nodo;

nodo = **registro**

dato: elemento;

izq, der: arbin

fregistro

acción insertar (**var** a: arbin; e: elemento)

si a = nulo **entonces**

reservar (a);

a^.dato:= e;

a^.izq:= nulo;

a^.der:= nulo

sino

si e < a^.dato **entonces**

insertar (a^.izq, e)

sino

insertar (a^.der, e)

fsi

fsi

facción

función está (a: arbin; e: elemento): booleano
si a = nulo **entonces retorna** (falso)
sino
en caso de
e < a^.dato: **retorna** (esta (a^.izq, e));
e = a^.dato: verdad;
e > a^.dato: **retorna** (esta (a^.der, e))
fcaso
fsi
ffunción

función min (a: arbin): elemento
si a^.izq = nulo **entonces retorna** (a^.dato)
sino retorna (min (a^.izq))
fsi
ffunción

función max (a: arbin): elemento
si a^.der = nulo **entonces retorna** (a^.dato)
sino retorna (max (a^.der))
fsi
ffunción

acción borrar (**var** a: arbin; e: elemento)
var
aux: arbin;
fvar

si a < > nulo **entonces**
en caso de
e < a^.dato: borrar (a^.izq, e);
e2 > a^.dato: borrar (a^.der, e);
e2 = a^.dato: **si** a^.izq = nulo **entonces**
aux:= a;
a:= a^.der;
liberar (aux)
sino
a^.dato:= max (a^.izq);
borrar (a^.izq, a^.dato)
fsi
fcaso
fsi
facción

fmódulo