

Implementación del TAD árbol binario

La manera más simple de representar un árbol binario es mediante registros que almacenan la información del nodo junto con dos punteros que lo enlazan con sus subárboles izquierdo y derecho.

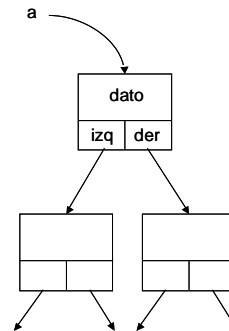
tipos

arbin = puntero a nodo;

nodo = registro

dato: elemento;
izq, der: arbin

fregistro



módulo árbolesBinarios

importa defTipoElemento

exporta

tipo arbin

acción creaVacío (var a: arbin)

acción plantar (e: elemento; ai, ad: arbin; var a: arbin)

función raíz (a: arbin): elemento

acción subIzq (a: arbin; var ai: arbin)

acción subDer (a: arbin; var ad: arbin)

función esVacío (a: arbin): booleano

función altura (a: arbin): entero

acción asignar (var nuevo: arbin; viejo: arbin)

acción liberarArbin (var a: arbin)

implementación

tipos

arbin = puntero a nodo;

nodo = registro

dato: elemento;
izq, der: arbin

fregistro

acción creaVacío (**var** a: arbin)

a:= nulo

facción

acción plantar (e: elemento; ai, ad: arbin; **var** a: arbin)

reservar (a);

a^.dato:= e;

a^.izq:= ai;

a^.der:= ad;

facción

función raíz (a: arbin): elemento

retorna (a^.dato)

ffunción

acción subIzq (a: arbin; **var** ai: arbin)

ai:= a^.izq

facción

acción subDer (a: arbin; **var** ad: arbin)

ad:= a^.der

facción

función esVacío (a: arbin): booleano

retorna (a = nulo)

ffunción

función altura (a: arbin): entero

en caso de

(a^.izq = nulo) \wedge (a^.der = nulo): **retorna** (1);

(a^.izq = nulo) \wedge (a^.der \neq nulo): **retorna** (1 + altura (a^.der));

(a^.izq \neq nulo) \wedge (a^.der = nulo): **retorna** (1 + altura (a^.izq));

(a^.izq \neq nulo) \wedge (a^.der \neq nulo): **retorna** (1 + max (altura (a^.izq), altura (a^.der)))

fcaso

ffunción

acción asignar (**var** nuevo: arbin; viejo: arbin)

var

ai, ad: arbin

fvar

si viejo = nulo **entonces**

nuevo:= nulo

sino

asignar (ai, viejo^.izq);

asignar (ad, viejo^.der);

reservar (nuevo);

nuevo^.dato:= viejo^.dato;

nuevo^.izq:= ai;

nuevo^.der:= ad

fsi

facción

acción liberarArbin (**var** a: arbin)

si a ≠ nulo **entonces**

liberarArbin (a^.izq);

liberarArbin (a^.der);

liberar (a);

a:= nulo

fsi

facción

fmódulo