Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Captura de pantalla de un celular con texto

Descripción generada automáticamente

TElementoAB

Lenguaje natural:

Dado un elemento comparar la etiqueta de ese nodo, con la del nodo donde estoy posicionado. En caso de ser menor lo asigno al hijo Izquierdo, sino al hijo derecho, en caso de ya existir no la asigno, para no tener repetidos.

Precondiciones.

El elemento a insertar debe ser distinto de nulo.

No puede existir ese elemento en el árbol.

Postcondiciones.

El elemento debe ser insertado a la izquierda del árbol si es menor y a la derecha si es mayor.

No pueden haber 2 elementos iguales.

Pseudocodigo

Diagrama

Descripción generada automáticamente

Insertar(ElementoAb elemento) Devuelve: booleano

COMIENZA

SI elemento.etiqueta < este.etiqueta ENTONCES

SI hijoIzq <> nulo ENTONCES

Devuelce hijoIzq.insertar(elemento)

SINO

hijoIzq 🡨 elemento

devolver verdadero

FINSI

SINO

SI hijoDer <> nulo ENTONCES

Devuelce hijoDer.insertar(elemento)

SINO

hijoDer 🡨 elemento

devolver verdadero

FINSI

SINO

Devolver falso

FINSI

FIN

TNodoArbolBB

Lenguaje natural:

Dado un ABB verificar que no este vacio, en caso de estarlo se asigna ese elemento al árbol siendo su raíz. En caso contrario se inserta utilizando la raíz, comparándolo consigo para saber si es menos, mayor o igual.

Precondiciones

Debe existir un árbol, puede tener elementos o estar vacio.

Poscondiciones

Al finalizar el árbol debe tener el elemento insertado.

En caso que el árbol inicial no estuviera vacio y fuera mayor la raíz, debe insertarse a la izquierda, en caso contrario insertarse a la derecha.

Pseudocodigo

Insertar(TElementoAB elemento) Devuelve verdadero o falso

Imagen que contiene Texto

Descripción generada automáticamenteArbolBB arbol

COMIENZA

SI Árbol. esVacio() ENTONCES

Raíz 🡨 elemento

Devuelve verdadero

SINO

Devuelve raíz.insertar(elemento)

FINSI

FIN

Texto, Aplicación, Escala de tiempo

Descripción generada automáticamente

TNodoArbolBB

Lenguaje natural:

Dado un ABB que puede estar vacio o contener elementos. Si la raíz es nulo devuelvo 0, sino recorro el árbol buscando que elementos no tienen hijos y los sumo.

Precondiciones:

Debe existir un ABB.

El ABB puede estar vacio o contener elementos.

Debe existir un contador inicializado en 0

Postcondiciones:

Debe devolver la cantidad de nodos hoja que tenga el árbol.

Pseucodigo

TArbolBB.cantidadHojas() devolver entero cantidad

TNodoArbolBB arbol

COMIENZO

SI árbol.esVacio() ENTONCES

Devolver 0.

SINO

Raíz.CantidadHojas()

FINSI

FIN

TElementoAB

Lenguaje Natural:

Evaluar si el nodo que estoy parado tiene ambos hijos nulos, en dicho caso aumentar el contado. En caso de no estarlo debo recorrer los hijos derecho e izquierdo preguntando lo mismo y hasta haber recorrido todo el árbol. Al final devolver la cantidad de nodos hoja del árbol.

Precondiciones

Debe existir un ABB.

El contador debe iniciar en 0

Postcondiciones:

El valor devuelto debe ser igual a la cantidad de nodos hojas del árbol.

El árbol no debe sufrir modificaciones.

Pseudo:

TElementoAB.cantidadHojas() devuelve entero cantidad

Contador 🡨0

COMIENZA

SI hijoIzq = nulo y hijoDer = nulo ENTONCES

Contador +1

FINSI

hijoIzq.cantidadHojas()

hijoDer.cantidadHoajs()

Devuelve contador

FIN