

TipoArbolBB.eliminar (unaEti de tipo etiqueta)

Comienzo

Si raíz <> nulo entonces

Raíz ← raíz.eliminar (unaEti)

Fin si

Fin

TipoNodoABB.eliminar (unaEti de tipo etiqueta): de TipoNodoABB

Comienzo

- | | |
|---|---|
| 1) Si unaEti < etiqueta entonces
Si hijoIzq <> nulo entonces
hijoIzq ← hijoIzq.eliminar(unaEti)
Finsi
retornar (this)
Finsi | <i>// si esta, está en el subárbol izquierdo</i>

<i>//actualiza el hijo, con el mismo u otro valor</i>

<i>//al padre le devuelve el mismo hijo</i> |
|
2) Si unaEti > etiqueta entonces
Si hijoDer <> nulo entonces
hijoDer ← hijoDer.eliminar(unaEti)
Finsi
retornar (this)
Finsi |
<i>// si esta, está en el subárbol derecho</i>

<i>//actualiza el hijo, con el mismo u otro valor</i>

<i>// al padre le devuelve el mismo hijo</i> |
|
3) retornar quitaElNodo |
<i>// esta, hay que eliminarlo</i>
<i>// al padre le devuelve el nuevo hijo</i> |

Fin

// Cuando encuentra el nodo a eliminar llama, por claridad, al método que hace el trabajo

TipoNodoABB.quitaElNodo: de TipoNodoABB

Comienzo

```
1) Si hijoIzq = nulo entonces // le falta el hijo izquierdo o es hoja
    retornar hijoDer           // puede retornar un nulo

2) Si hijoDer = nulo entonces // le falta el hijo derecho
    retornar hijoIzq

3) // es un nodo completo
    elHijo ← hijoIzq           // va al subárbol izquierdo
    elPadre ← this
    mientras elHijo.hijoDer <> nulo hacer
        elPadre ← elHijo
        elHijo ← elHijo.hijoDer
    fin mientras              // elHijo es el mas a la derecha del subárbol izquierdo

Si elPadre <> this entonces
    elPadre.hijoDer ← elHijo.hijoIzq
    elHijo.hijoIzq ← hijoIzq
Finsi

elHijo.hijoDer ← hijoDer
retornar elHijo              // elHijo quedara en lugar de this
```

Fin