

[illegible]

Vitoria-Gasteiz, 21 de diciembre de 2011

2. Diseño

Casos triviales

Si esVacio (A) → devolver new Contexto(0, 0)

Si esHoja (A) y no(esDerecho) → devolver new Contexto(raiz(A), 0)

Si esHoja (A) y esDerecho → devolver new Contexto(2*raiz(A), 0)

Caso General

Si No esHoja(A) →

costeA = (esDerecho? 2 : 1) * raiz(A);

contextoIzq = *longitudCaminoCosteMaximo1* (subarbolIzq(A), true);

contextoDch = *longitudCaminoCosteMaximo1* (subarbolDch(A), false);

Si (contextoIzq. coste > contextoDch.coste) **entonces**

coste = contextoIzq.coste + costeA;

longitud = contextoIzq.longitud + 1;

si no (contextoIzq. coste < contextoDch.coste) **entonces**

coste = contextoDch. coste + costeA;

longitud = contextoDch.longitud + 1;

si no

coste = contextoDch. coste + costeA;

longitud = maximo(contextoIzq.longitud, contextoDch.longitud) + 1;

fin si

contexto = new Contexto(coste, longitud);

devolver contexto;

Vitoria-Gasteiz, 21 de diciembre de 2011

3-Implementación

```
class BinTreeItr<T> {  
  
...  
  
public int longitudCaminoCosteMaximo() {  
    Contexto contexto = longitudCaminoCosteMaximo1 (  
                                                (BTNode<Integer> bTree.root, false);  
    return contexto.longitud;  
}  
  
private Contexto longitudCaminoCosteMaximo1(BTNode<? extends Integer> A,  
                                             boolean esDerecho) {  
    int coste, longitud;  
  
    if (A==null)  
        return new Contexto(0, 0);  
    else if ((A.left==null) && (A.right==null) ) { // A es hoja  
        coste = (esDerecho? 2 : 1) * A.content.intValue();  
        return new Contexto(coste, 0);  
    }  
    else {  
        int costeA = (esDerecho? 2 : 1) * A.content.intValue();  
        Contexto contextoIzq = longitudCaminoCosteMaximo1(A.left, false);  
        Contexto contextoDch = longitudCaminoCosteMaximo1(A.right, true);  
        if (contextoIzq.coste > contextoDch.coste){  
            coste = contextoIzq.coste + costeA;  
            longitud = contextoIzq.longitud + 1;  
        }  
        else if (contextoIzq.coste < contextoDch.coste){  
            coste = contextoDch.coste + costeA;  
            longitud = contextoDch.longitud + 1;  
        }  
        else { //el coste máximo es el mismo  
            coste = contextoDch.coste + costeA;  
            longitud = Math.max(contextoIzq.longitud, contextoDch.longitud)+1;  
        }  
        Contexto contexto = new Contexto(coste, longitud);  
        return contexto;  
    }  
}  
}  
  
public class Contexto {  
    public int coste, longitud;  
    public Contexto(int coste, int longitud) {  
        this.coste = coste;  
        this.longitud = longitud;  
    }  
}
```