

Lenguaje Natural:

Si el árbol es vacío devolvemos verdadero, ya que la altura es 0, entonces cumpliría con AVL, sino empezamos a recorrer el árbol utilizando una llamada recursiva con un vector boolean pasado por parámetro, y devolvemos al final el valor 0 de dicho vector. En la recorrida vamos calculando las alturas para cada sub árbol, si los hijos son no nulos, luego de recorrer ambos hijos, verificamos si su diferencia de las alturas en valor absoluto, es mayor que 1, en caso de serlo al vector cumple lo ponemos en Falso, y sino en verdadero ya que cumple con AVL. Luego de esto devolvemos la mayor altura + 1.

Precondiciones:

- Si es vacío cumple con AVL

Pos condiciones:

- El estado del árbol no se toca.
- El estado del vector cumple es verdadero si cumple con AVL
- En el TElementoAB el método debería de devolver la altura correcta

Seudocódigo:

En TArbolBB

De tipo boolean cumpleAVL ():

Comienzo

```
Si esVacio() entonces    // O(1)
    Devolver Verdadero    // O(1)
Fin Si
Boolean[] cumple= { Verdadero }    // O(1)
raiz.cumpleAVL(cumple)
Devolver cumple[0]    // O(1)
```

Fin

En TElementoAB

De tipo entero cumpleAVL (boolean[] cumple):

Comienzo

```
Entero subArbolIzq = -1    // O(1)
Entero subArbolDer = -1    // O(1)
Si hijoIzq <> nulo Entonces    // O(1)
    subArbolIzq = hijoIzq.cumpleAVL(cumple)
Fin Si
Si hijoDer <> nulo Y cumple[0] Entonces    // O(1)
    subArbolDer = hijoDer.cumpleAVL(cumple)
Fin Si
```

Si  $\text{absoluto}(\text{subArbolIzq} - \text{subArbolDer}) > 1$  Entonces // O(1)

$\text{cumple}[0] = \text{Falso}$  // O(1)

Fin Si

Si  $\text{subArbolIzq} > \text{subArbolDer}$  entonces // O(1)

    Devolver  $1 + \text{subArbolIzq}$  // O(1)

Fin Si

Devolver  $1 + \text{subArbolDer}$  // O(1)

Fin

El orden total es de orden  $O(N)$ , siendo N la cantidad de elementos del árbol.

Casos de prueba:

- El tamaño del árbol sea el mismo
- Que la altura del árbol sea correcta
- El estado de cumple en su índice 0 tiene si el árbol cumple con AVL o no.