

Ejercicio 9.- (Examen del Grado en Ingeniería Informática, Enero 2012)

Llamaremos a un árbol general de naturales “creciente” en cada nivel del árbol, la cantidad de nodos que hay en ese nivel es igual al valor del nivel más uno; es decir, el nivel 0 tiene exactamente un nodo, el nivel 1 tiene exactamente dos nodos, el nivel k tiene exactamente $k + 1$ nodos. Se pide:

- Especificar completamente el TAD árbol general,
- Comprobar si un árbol general es “creciente”,
- Buscar el nodo con mayor cantidad de hijos de un árbol creciente.

Necesitaremos una función auxiliar que cuente el total de nodos de un nivel dado k :

func nodos_nivel_k (a:árbol, k:natural) **dev** n:natural

si $k=0$ **entonces** $n \leftarrow 1$

sino $n \leftarrow \text{nodos_nivel_k_b}(\text{hijos}(a), k-1)$

finsi

finfunc

func nodos_nivel_k_b(b:bosque, k:natural) **dev** n:natural

si vacío?(b) **entonces** $n \leftarrow 0$

sino si $k=0$ **entonces** $n \leftarrow \text{long}(b)$ { tamaño del bosque }

sino

$n \leftarrow \text{nodos_nivel_k}(\text{primero}(b), k) + \text{nodos_nivel_k_b}(\text{resto}(b), k)$

finsi

finfunc

func creciente (a:árbol) **dev** b:boolean

 creciente_desde_k(a,1)

finfunc

func creciente_desde_k (a: árbol,k:natural) **dev** b:boolean

```

si nodos_nivel_k(a, k) = 0 entonces b ← T
sino si nodos_nivel_k(a, k) != k+1 entonces b ← F
    sino b ← creciente_desde_k(a, k+1)
finsi
finfunc

o tb. {versión iterativa}
func creciente (a:árbol) dev b:boolean
    k ← 0 es_creciente ← T
mientras nodos_nivel_k(a, k) != 0 ∧ es_creciente hacer
    si nodos_nivel_k(a, k) != k+1 entonces es_creciente ← F
    sino k ← k+1
    finsi
finmientras
finfunc

```