

# Algoritmos y Estructuras de Datos.

## 2do Parcial. Recup. Especial. [18 de junio de 2003]

**[Ej. 1] [Primitivas]** Escribir las funciones del TAD ARBOL BINARIO listadas a continuación, con *celdas enlazadas por punteros ó cursores*, a saber: PADRE( $n, A$ ), HIJO\_IZQ( $n, A$ ), HIJO\_DER( $n, A$ ), ETIQUETA( $n, A$ ), CREA2( $v, A1, A2$ ) y ANULA( $A$ ). **Escribir todos los tipos, definiciones, funciones y procedimientos auxiliares necesarios.**

**[Ej. 2] [Programación ]**

- (a) **[MAXCOTA]** Escribir una función `function MAXCOTA(n:nodo; A:arbol; cota:integer):integer`; que retorna el máximo de las etiquetas de un árbol binario tales que son menores o iguales que la cota  $c$ . Por ejemplo, si las etiquetas de un árbol  $A$  son  $\{1, 3, 7, 4, 2, 10, 13\}$  y  $cota=8$ , entonces `MAXCOTA(raiz(A), A, 8)` debe retornar 7. Usar las primitivas del TAD ARBOL BINARIO: `HIJO_IZQ(n, A)`, `HIJO_DER(n, A)`, `ETIQUETA(n, A)` y `raiz(A)`.
- (b) **[INCLUIDO]** Escribir una función `function INCLUIDO(nb, na: nodo; B, A: arbol) : boolean`, la cual retorna verdadero si la *estructura* del subárbol del nodo  $nb$  en el árbol ordenado orientado  $B$  está incluida dentro de la del subárbol del nodo  $na$  en el árbol  $A$ , independientemente de las etiquetas de los nodos correspondientes. Definiéndolo en forma recursiva, el subárbol del nodo  $nb$  está incluido en el de otro nodo  $na$  si:
- $nb = \Lambda$ , o
  - $nb \neq \Lambda$ ,  $na \neq \Lambda$  y los subárboles de los hijos de  $nb$  están incluidos en los subárboles de los hijos de  $na$ . (Notar la *recursividad* de la definición).
- Utilizar las funciones `PADRE (n, A)`, `HIJO_MAS_IZQUIERDO (n, A)`, `HERMANO_DERECHO (n, A)`, y `ETIQUETA (n, A)`.
- (c) **[CUENTA\_COTA]** Escribir una función `function CUENTA_COTA(n:nodo; A:arbol; cota:integer):integer` que retorna el numero de nodos de un árbol orientado que tienen etiqueta menor o igual que cota. Utilizar las funciones `PADRE (n, A)`, `HIJO_MAS_IZQUIERDO (n, A)`, `HERMANO_DERECHO (n, A)`, y `ETIQUETA (n, A)`.

**[Ej. 3] [Operativos]**

- (a) **[Reconstruir árbol ]** Dibujar el árbol ordenado orientado cuyos nodos, listados en orden previo y posterior son
- $ORD\_PRE = \{Z, T, U, V, X, Y, W, A, B\}$ ,
  - $ORD\_POST = \{X, Y, V, A, B, W, U, T, Z\}$ .
- (b) **[Ubicar nodos]** Dado el árbol siguiente
- i. Ubicar los nodos que estén a la izquierda de  $A$  y a la derecha de  $I$
  - ii. Ubicar los nodos que estén a la derecha de  $D$  y sea descendiente de  $J$
- (c) **[Árbol de expresiones]** Dada la expresión siguiente

$$(a + d * e - f/g) * (h/p - r) \tag{1}$$

encontrar el árbol binario correspondiente. Escribir la expresión en notación prefija y postfija.