

Algoritmos y Estructuras de Datos.

Parcial 2. Tema 1c. [28 de Mayo de 2002]

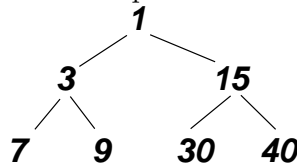
Ej. 1.- Dibujar el árbol ordenado orientado cuyos nodos, listados en orden previo y posterior son

- $ORD_PRE = \{W, B, R, S, Q, Z, A, B\}$,
- $ORD_POST = \{R, S, B, Q, A, B, Z, W\}$.

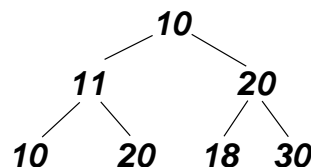
Ej. 2.- Escribir las funciones primitivas del TAD ARBOL ORDENADO ORIENTADO listadas a continuación, con celdas enlazadas por punteros o cursores:

- (a) PADRE(n, A)
- (b) HIJO_MAS_IZQ(n, A)
- (c) HERMANO_DER(n, A)
- (d) ETIQUETA(n, A)
- (e) CREA2($v, A1, A2$)
- (f) ANULA(A)

Ej. 3.- Se dice que un árbol binario es “*parcialmente ordenado*” (PO) si la etiqueta de cada nodo es menor o igual que las etiquetas de sus hijos. Así por ejemplo de los dos árboles siguientes, el de la izquierda verifica la condición, mientras que el de la derecha no.



SI es PO.



NO es PO.

Escribir una función `VERIFICA_PO(n:nodo; A: Arbol) : boolean` que retorna `true` si el subárbol que cuelga del nodo `n` es parcialmente ordenado y `false` en caso contrario. Usar las primitivas del TAD ARBOL BINARIO: `HIJO_IZQ(n,A)`, `HIJO_DER(n,A)`, `ETIQUETA(n,A)`.

Ej. 4.- Árboles de Huffman: Dados los caracteres siguientes con sus correspondientes probabilidades, contruir el código binario y encodar la palabra **CRISIS**

$P(c) = 0.2, P(r) = 0.2, P(s) = 0.1, P(i) = 0.3,$
 $P(z) = 0.05, P(u) = 0.05, P(v) = 0.05, P(t) = 0.05.$
Calcular la longitud promedio del código obtenido.

Ej. 5.- Insertar los números 27, 10, 37, 47, 22, 57, 67, 31 en una tabla de dispersión cerrada con $B = 10$ cubetas, con función de dispersión $h(x) = x \bmod 10$ y estrategia de redispersión lineal.