

Algoritmos y Estructuras de Datos.

Examen Final. [3 de Octubre de 2002]

- Ej. 1.-** Escribir las funciones primitivas del TAD ARBOL BINARIO listadas a continuación, con celdas enlazadas por punteros o cursores: PADRE(n, A), HIJO_IZQ(n, A), HIJO_DER(n, A), ETIQUETA(n, A), CREA2($v, A1, A2$) y ANULA(A).
- Ej. 2.-** Escribir un procedimiento `procedure KRONECKER(LA, LB: lista; var L: lista);` que, dadas dos listas $LA = \{a_1, a_2, \dots, a_N\}$ y $LB = \{b_1, b_2, \dots, b_M\}$ retorna otra lista L con $N \times M$ valores de la siguiente forma:

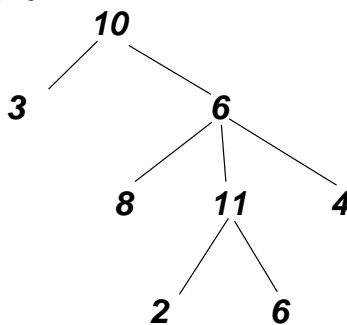
$$\begin{aligned}
 L = \{ & a_1b_1, a_1b_2, \dots, a_1b_M, \\
 & a_2b_1, a_2b_2, \dots, a_2b_M, \\
 & \dots \\
 & a_Nb_1, a_Nb_2, \dots, a_Nb_M \}
 \end{aligned}
 \tag{1}$$

Por ejemplo, si $LA = \{1, 3, 5, 0\}$ y $LB = \{1, 2, 3, 4\}$ entonces $KRONECKER(LA, LB, L)$ debe hacer que

$$L = \{1, 2, 3, 4, 3, 6, 9, 12, 5, 10, 15, 20, 0, 0, 0, 0\}. \tag{2}$$

Usar las siguientes primitivas del TAD LISTA: INSERTA(x, p, L), RECUPERA(p, L), SUPRIME(p, L), SIGUIENTE(p, L), ANULA(L), PRIMERO(L), y FIN(L). No usar ninguna estructura auxiliar.

- Ej. 3.-** 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 $raiz(A)$.
- Ej. 4.-** Escribir una función `function MAXHOJA(n:nodo; A:arbol):integer;` que retorna el máximo de las etiquetas de las hojas un árbol ordenado orientado. Por ejemplo, en el siguiente árbol debe retornar 8.



Usar las primitivas del TAD ARBOL ORDENADO ORIENTADO: HIJO_MAS_IZQ(n, A),
HERMANO_DER(n, A), ETIQUETA(n, A).

Ej. 5.- [LIBRES] Ejercicios operativos:

- (a) **Árboles:** Dibujar el árbol ordenado orientado cuyos nodos, listados en orden previo y posterior son
- $ORD_PRE = \{C, Q, T, R, S, U, A, L\}$,
 - $ORD_POST = \{T, Q, S, U, L, A, R, C\}$.
- (b) **Clasificación por montículos:** Dados los enteros $\{15, 7, 3, 8, 4, 10, 2\}$ ordenarlos por el método de “montículos” (“*heap-sort*”). Mostrar el montículo (minimal) antes y después de cada inserción/supresión.

Ej. 6.- [LIBRES] Preguntas: [Responder según el sistema “multiple choice”, es decir marcar con una cruz el casillero apropiado. **Atención:** Algunas respuestas son intencionalmente “descabelladas” y tienen puntajes **negativos!!**]

- (a) Dadas las funciones $T_1(n) = 2n^3 + \log n$, $T_2(n) = 4 + \log n$, $T_3(n) = 2^n + n!$ y $T_4(n) = 4^n + n^2$ decir cuál de los siguientes ordenamientos es el correcto
- ☐ $T_1 < T_2 < T_3 < T_4$
- ☐ $T_4 < T_3 < T_2 < T_1$
- ☐ $T_2 < T_1 < T_4 < T_3$
- ☐ $T_3 < T_4 < T_1 < T_2$
- (b) ¿Cuál de los siguientes algoritmos de clasificación es el más rápido en el caso promedio?
- ☐ Burbuja (“*Bubble-sort*”)
- ☐ Clasificación rápida (“*Quick-sort*”)
- ☐ Clasificación óptima (“*Holy grail sort*”)
- ☐ Selección
- (c) Una ventaja del método de clasificación por selección, en comparación con otros algoritmos lentos, es que realiza sólo n intercambios...
- ☐ ... a veces.
- ☐ ... cuando el vector está ordenado.
- ☐ ... siempre.
- ☐ ... cuando el vector está desordenado.