

## Algoritmos y Estructuras de Datos.

### Parcial 1. Tema 1a. [16 de abril de 2002]

**Ej. 1.-** Ordenar por velocidad de crecimiento **de menor a mayor** las siguientes funciones

- (a)  $T_1(n) = 4.3n^2 + 2n + 0.3 \log(n)$ ,  $T_2(n) = 3.5 \times 2^n + 5n^3 + 2\sqrt{n}$ ,  
 $T_3(n) = 0.3n$ ,  $T_4(n) = 23$ .  
(b)  $T_5(n) = 2n$ ,  $T_6(n) = \sqrt{n}$ ,  $T_7(n) = 0.3 \log n + 5$   
(c)  $T_8(n) = n^3 + \sqrt{n} + 2^n$ ,  $T_9(n) = 0.3n + 5n^3$ ,  $T_{10}(n) = 23 + 2n^2$ ,  $T_{11}(n) = n!$

**Ej. 2.-** Escribir las funciones primitivas del TAD Lista con celdas simplemente enlazadas por punteros. Es decir, implementar en Pascal los siguientes procedimientos/funciones:

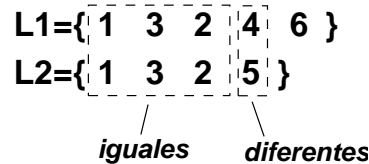
- (a) `INSERTA(x,p,L)`,  
(b) `LOCALIZA(x,L)`,  
(c) `RECUPERA(p,L)`,  
(d) `SUPRIME(p,L)`,  
(e) `SIGUIENTE(p,L)`,  
(f) `ANULA(L)`,  
(g) `PRIMERO(L)`, y  
(h) `FIN(L)`.

[Nota: Se recomienda utilizar celda de encabezamiento. Puede usarse puntero a la última celda o no.]

**Ej. 3.-** Dadas dos listas de enteros L1 y L2 escribir una función

```
function LEXICORD(L1,L2:lista) : boolean;
```

que retorna `true` si la lista L1 es mayor que la L2 en sentido “*lexicográfico*”, y `false` en caso contrario. El orden lexicográfico es el orden alfabético usado para ordenar las palabras en el diccionario, si consideramos que la lista de enteros es una “*palabra*” cuyas “*letras*” son cada uno de los enteros de la lista. Luego, se van comparando los números en las posiciones correspondientes hasta encontrar uno diferente. La lista que tiene el número ( “*letra*”) mayor es la lista “*mayor*” en el sentido lexicográfico. Así, por ejemplo, si  $L1=(1 \ 3 \ 2 \ 4 \ 6)$ ,  $L2=(1 \ 3 \ 2 \ 5)$ , entonces  $L2 > L1$  ya que los primeros elementos diferentes son 5 y 4 en la cuarta posición, de los cuales es mayor el 5 que está en L2 (ver figura).



Por lo tanto  $\text{LEXICORD}(L1, L2)$  debe retornar **false**. Además si una lista esta totalmente contenida en el principio de la otra, es “*mayor*” la lista más larga. Por ejemplo, si  $L1 = (1 \ 3 \ 3 \ 4 \ 5)$   $L2 = (1 \ 3 \ 3)$ , entonces  $L1 > L2$  y  $\text{LEXICORD}(L1, L2)$  retorna **true**. Dos listas son iguales, si y sólo si tienen la misma cantidad de elementos, y los elementos en posiciones correspondientes son iguales. Por ejemplo, si  $L1 = (1 \ 3 \ 3 \ 4)$  y  $L2 = (1 \ 3 \ 3 \ 4)$ , entonces  $L1 = L2$  y  $\text{LEXICORD}(L1, L2)$  retorna **false**.

El orden lexicográfico entre dos listas  $L1$  y  $L2$  puede definirse, en forma más precisa, de la siguiente manera,

- (a) Si las dos listas son vacías, entonces  $L1 = L2$  (valor de retorno **false**)
- (b) Si una de las listas es vacía (digamos  $L2$ ) y la otra ( $L1$ ) no lo es, entonces  $L1 > L2$  (retorna **true**). Si la que esta vacía es  $L1$  entonces  $L1 < L2$  y retorna **false**.
- (c) Si las dos listas no son vacías, y los primeros elementos son diferentes (digamos  $a_1$  en  $L1$  y  $b_1$  en  $L2$ ), entonces el valor de retorno es el valor lógico de  $a_1 > b_1$ .
- (d) Finalmente, si los dos primeros elementos son iguales, entonces el valor de retorno es el que corresponde a las listas que se obtienen eliminando los primeros elementos.

Utilizar las primitivas del **TAD LISTA**:  $\text{INSERTA}(x, p, L)$ ,  $\text{RECUPERA}(p, L)$ ,  $\text{SUPRIME}(p, L)$ ,  $\text{SIGUIENTE}(p, L)$ ,  $\text{ANULA}(L)$ ,  $\text{PRIMERO}(L)$ , y  $\text{FIN}(L)$ .

**Ej. 4.-** Escribir los siguientes procedimientos/funciones

- (a) Escribir un procedimiento  $\text{ROTACION}(\text{var } C: \text{cola})$  que saca una cierta cantidad de enteros del frente de la cola  $C$  y los vuelve a insertar en fin de cola, de tal manera que quede en el frente de cola un número par. Por ejemplo, si  $C = \{1, 3, 5, 2, 4\}$  entonces, después de  $\text{ROTACION}(C)$  debe quedar  $C = \{2, 4, 1, 3, 5\}$   
 Utilizar las primitivas del **TAD COLA**:  $\text{ANULA}(C)$ ,  $\text{PONE\_EN\_COLA}(x, C)$ ,  $\text{QUITA\_DE\_COLA}(C)$ ,  $\text{VACIA}(C)$ , y  $\text{FRENTE\_DE\_COLA}(C)$ .
- (b) Escribir un procedimiento  $\text{SACA\_FONDO}(\text{var } P: \text{pila})$ ; que elimina el último elemento de una pila  $P$  dejando los demás inalterados, usando exclusivamente una pila auxiliar.  
 Utilizar las primitivas del **TAD PILA**:  $\text{ANULA}(P)$ ,  $\text{METE}(x, P)$ ,  $\text{SACA}(P)$ ,  $\text{TOPE}(P)$  y  $\text{VACIA}(P)$ .