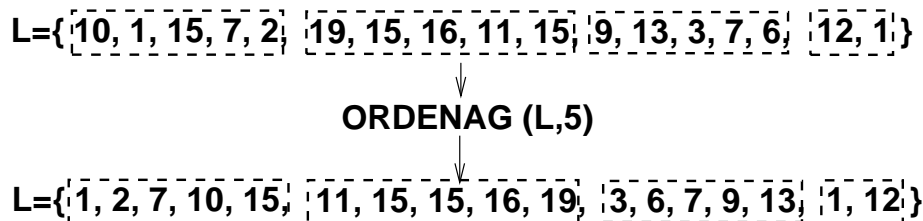


## Algoritmos y Estructuras de Datos.

### Examen Final. [5 de Diciembre de 2002]

- Ej. 1.-** Escribir las funciones primitivas del TAD ARBOL ORDENADO ORIENTADO listadas a continuación, con celdas enlazadas por punteros o cursores: PADRE( $n, A$ ), HIJO\_MAS\_IZQ( $n, A$ ), HERMANO\_DER( $n, A$ ), ETIQUETA( $n, A$ ), CREA2( $v, A1, A2$ ) y ANULA( $A$ ).
- Ej. 2.-** Escribir un procedimiento `procedure ORDENAG(var L: lista; m:integer);` que, dada una lista  $L$  va ordenando sus elementos de a grupos de  $m$  elementos. Por ejemplo si  $m = 5$  ORDENAG ordena los primeros 5 elementos entre sí, después los siguientes 5 elementos, y así siguiendo. Si la longitud  $N$  de la lista no es un múltiplo exacto de  $m$  elementos, entonces los últimos  $N \bmod m$  elementos *también* deben ser ordenados entre sí. En la figura se muestra el resultado de aplicar ORDENAG( $L, 5$ ) a una lista de 17 elementos.



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( $L$ ). Sugerencia: Usar una lista auxiliar. Para ordenar los elementos dentro de esta lista auxiliar puede usarse cualquier algoritmo de ordenamiento. Se sugiere usar inserción o similar.

- Ej. 3.-** Escribir una función `function SUMAPAR(n: nodo; A: arbol) : integer` que retorna la suma de las etiquetas pares de un árbol binario. Usar las primitivas del TAD ARBOL BINARIO: HIJO\_IZQ( $n, A$ ), HIJO\_DER( $n, A$ ), ETIQUETA( $n, A$ ).
- Ej. 4.-** Escribir una función `function MIN(n:nodo; A:arbol):integer;` que retorna el mínimo de las etiquetas de un árbol ordenado orientado. 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 = { $Q, R, S, L, W, M, Z, T$ },
  - ORD\_POST = { $R, W, L, Z, M, S, T, Q$ }.

- (b) **[LIBRES]** Dados los enteros  $\{12, 11, 14, 8, 16, 13, 9, 7, 15, 21, 4\}$  insertarlos, en ese orden, en un “árbol binario de búsqueda”. Mostrar las operaciones necesarias para eliminar los elementos 12, 9 y 13.

**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  $T_1(n) = 5n + \log n$ ,  $T_2(n) = 4n^2 + \sqrt{n}$ ,  $T_3(n) = 2^n + n!$  y  $T_4(n) = \sqrt{n} + \log n$  decir cuál de los siguientes ordenamientos es el correcto

☐  $T_4 < T_3 < T_2 < T_1$

☐  $T_2 < T_1 < T_4 < T_3$

☐  $T_4 < T_1 < T_2 < T_3$

☐  $T_3 < T_4 < T_1 < T_2$

- (b) ¿Cuál de los siguientes algoritmos de clasificación es  $O(n \log n)$  en el caso promedio?

☐ Clasificación secuencial (“Sequential sort”)

☐ Clasificación rápida (“Quick-sort”)

☐ Selección

☐ Burbuja (“Bubble-sort”)

- (c) La búsqueda *exhaustiva* del camino de longitud mínima para el “Problema del Agente Viajante” (PAV) tiene un tiempo de ejecución... ( $n$  es el número de ciudades)

☐  $O(n!)$

☐  $O(1)$

☐  $O(n)$

☐  $O(n^2)$

- (d) Una de las desventajas de implementar el TAD LISTA con arreglos es que las inserciones/supresiones en el medio de la lista son...

☐ ...  $O(1)$ .

☐ ...  $O(n^2)$ .

☐ ...  $O(n)$ .

☐ ...  $O(\log n)$ .