

Algoritmos y Estructuras de Datos.

1er Parcial. Tema: **1c.** [22 de Abril de 2003]

[Ej. 1] [Tiempos de ejecución (10 puntos)] Dadas las funciones

- $T_1(n) = 0.3 \log_2(n) + 3n^2$
- $T_2(n) = n! + 2^n$
- $T_3(n) = 3^n + n^2$
- $T_4(n) = 0.5\sqrt{n} + 2n^{0.9}$

decir cuál de los siguientes ordenamientos es el correcto

- ☐ $T_3 < T_4 < T_2 < T_1$
- ☐ $T_2 < T_3 < T_4 < T_1$
- ☐ $T_4 < T_1 < T_3 < T_2$
- ☐ $T_3 < T_4 < T_1 < T_2$

[Ej. 2] [Primitivas (15 puntos)] Escribir las funciones primitivas del TAD Lista con celdas simplemente enlazadas por *cursores*. Es decir, implementar en Pascal los siguientes procedimientos/funciones: `INSERTA(x,p,L)`, `LOCALIZA(x,L)`, `RECUPERA(p,L)`, `SUPRIME(p,L)`, `SIGUIENTE(p,L)`, `ANULA(L)`, `PRIMERO(L)`, y `FIN(L)`. [Nota: Se recomienda utilizar celda de encabezamiento. Puede usarse puntero a la última celda o no.]

[Ej. 3] [Programación (total = 45 puntos)] Dada una secuencia de números $\{a_1, a_2, \dots, a_n\}$, vamos a decir que su “*máxima desviación*”, es la máxima diferencia (en valor absoluto) entre todos sus números: $\max_dev(a_1, a_2, \dots, a_n) = (\max_{j=1}^n a_j) - (\min_{j=1}^n a_j)$.

- (a) [35 puntos] Escribir una función “`function MAX_DEV_M(L:list; m:integer) : integer;`” que retorna el máximo de las máximas desviaciones de las subsecuencias de L de longitud m, es decir

$$\max_dev_m(L) = \max\{\max_dev(a_1, a_2, \dots, a_m), \max_dev(a_2, a_3, \dots, a_{m+1}), \max_dev(a_3, \dots, a_{m+2}), \dots, \max_dev(a_{n-m+1}, \dots, a_n)\} \quad (1)$$

Por ejemplo, si $L=(1,3,5,4,3,5)$, entonces `MAX_DEV_N(L,3)` debe retornar 4 ya que la máxima desviación se da en la primera subsecuencia $(1,3,5)$ y es 4. Se sugiere el siguiente algoritmo, para cada posición p en la lista hallar la máxima desviación de los m elementos siguientes (incluyendo a p). Hallar la máxima de estas desviaciones. Utilizar las primitivas del **TAD LISTA**: `INSERTA(x,p,L)`, `RECUPERA(p,L)`, `SUPRIME(p,L)`, `SIGUIENTE(p,L)`, `ANULA(L)`, `PRIMERO(L)`, y `FIN(L)`.

- (b) [5 puntos] Cual es el tiempo de ejecución, en el peor caso, si $m=2$, como función de n,
- (c) [5 puntos] Cual es el tiempo de ejecución en el peor caso, si $m=n/2$ (asumimos que n es par), como función de n.

[Ej. 4] [Programación básica de pilas y colas (total = 20 puntos)] Escribir los siguientes procedimientos/funciones

- (a) [10 puntos] Escribir una función “function MAXPILA(C:pila) : integer” que retorna el máximo de los elementos de una pila usando una pila auxiliar. Finalmente la pila debe quedar en el mismo estado que originalmente. Usar las primitivas del TAD PILA: ANULA(P), METE(x,P), SACA(P), TOPE(P) y VACIA(P).
- (b) [10 puntos] Escribir un procedimiento “procedure X2(var C: cola);” que invierte los elementos de una cola de a 2, es decir el primero con el 2do, el 3ro con el 4to, etc... usando una cola auxiliar. Si la cola tiene un número impar de elementos entonces el último queda inalterado. Por ejemplo, si $C=(frente=2,4,3,2,5,6,7)$ entonces después de $X2(C)$; debe quedar $C=(frente=4,2,2,3,6,5,7)$.

[Ej. 5] [Preguntas (total = 10 puntos, 2.5puntos por pregunta)] 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) El tiempo de ejecución de la función ANTERIOR para listas *simplemente* enlazadas es ... (n es el número de elementos en la lista)
- ☐ ... $O(n^2)$
- ☐ ... $O(1)$
- ☐ ... $O(\log n)$
- ☐ ... $O(n)$
- (b) El tiempo de ejecución de la función PONE_EN_COLA para colas implementadas por arreglos circulares es ... (n es el número de elementos en la cola)
- ☐ ... $O(1)$
- ☐ ... $O(\log n)$
- ☐ ... $O(n^2)$
- ☐ ... $O(n)$
- (c) La desventaja de la implementación del TAD CORRESPONDENCIA por arreglos es que ... (N_d es el número de elementos en el dominio, n es el número de elementos del dominio que tienen valores asignados).
- ☐ ... el tamaño de la correspondencia es $O(N_d)$.
- ☐ ... el tamaño de la correspondencia es $O(n)$.
- ☐ ... el tiempo de ejecución de CALCULA(M,d,r) es $O(N_d)$.
- ☐ ... el tiempo de ejecución de CALCULA(M,d,r) es $O(n)$.
- (d) El tiempo de ejecución de la función CALCULA(M,d,r) para el TAD CORRESPONDENCIA implementado por listas, en el peor caso es... (N_d es el número de elementos en el dominio, n es el número de elementos del dominio que tienen valores asignados).
- ☐ ... $O(n)$
- ☐ ... $O(N_d)$
- ☐ ... $O(n^2)$
- ☐ ... $O(1)$