

Algoritmos y Estructuras de Datos.

1er Parcial. Tema: **2c.** [22 de Abril de 2003]

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

- $T_1(n) = 2^n + n^2$
- $T_2(n) = 3^n + n^3$
- $T_3(n) = \sqrt{n} + \log n$
- $T_4(n) = \sqrt{n} + n!$

decir cuál de los siguientes ordenamientos es el correcto

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

[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 un procedimiento “`procedure SUAVIZA_M(var L:lista; m,maxdif:integer) : integer;`” que elimina la mínima cantidad de elementos de L de tal manera que la máxima desviación de una subsecuencia de m elementos consecutivos es maxdif. Por ejemplo, si $L=(1,3,5,4,2,3,7,4)$ entonces `SUAVIZA_M(L,3,3)` debe retornar $L=(1,3,4,2,3,4)$, habiéndose eliminado los elementos 5 y 7. Se sugiere el siguiente algoritmo, para cada posición p en la lista recorrer los m-1 elementos siguientes a p, removiendo aquellos elementos que tienen una diferencia con el elemento p mayor a maxdif. 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 un procedimiento “`procedure DEJAPAR(var P:pila)`” que elimina de la pila P todos los elementos impares usando una pila auxiliar. Los elementos deben quedar en el mismo orden en el que estaban. Por ejemplo, si $P=(tope=6,1,2,4,3,5,6)$ entonces después de `DEJAPAR(P)`, debe quedar $P=(tope=6,2,4,6)$. Usar las primitivas del TAD PILA: `ANULA(P)`, `METE(x,P)`, `SACA(P)`, `TOPE(P)` y `VACIA(P)`.

Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]

- (b) [10 puntos] Escribir una función “function MAXCOLA(C:pila) : integer” que retorna el máximo de los elementos de una cola usando una cola auxiliar. Finalmente la cola debe quedar en el mismo estado que originalmente. Utilizar las primitivas del **TAD COLA**: ANULA(C), PONE_EN_COLA(x,C), QUITA_DE_COLA(C), VACIA(C), y FRENTE_DE_COLA(C).

[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 LOCALIZA para listas *simplemente* enlazadas es $O(1)$... (n es el número de elementos en la lista)
- ☐ ... siempre.
- ☐ ... en el mejor caso.
- ☐ ... en el peor caso.
- ☐ ... cuando el elemento no está en la lista.
- (b) El tiempo de ejecución de la función CALCULA(M,d,r) para el TAD CORRESPONDENCIA implementado por arreglos es... (N_d es el número de elementos en el dominio, n es el número de elementos del dominio que tienen valores asignados).
- ☐ ... $O(N_d)$
- ☐ ... $O(n)$
- ☐ ... $O(1)$
- ☐ ... $O(n^2)$
- (c) El requerimiento de memoria para el TAD CORRESPONDENCIA implementado por listas es... (N_d es el número de elementos en el dominio, n es el número de elementos del dominio que tienen valores asignados).
- ☐ ... $O(n)$
- ☐ ... $O(N_d^2)$
- ☐ ... $O(1)$
- ☐ ... $O(N_d)$
- (d) ¿Cuál de las siguientes expresiones booleanas indica si una lista L esta vacía?
- ☐ ... PRIMERO(L)+1 = FIN(L)
- ☐ ... SIGUIENTE(PRIMERO(L),L) = FIN(L)
- ☐ ... PRIMERO(L) = FIN(L)
- ☐ ... PRIMERO(L) <> FIN(L)