

Algoritmos y Estructuras de Datos.

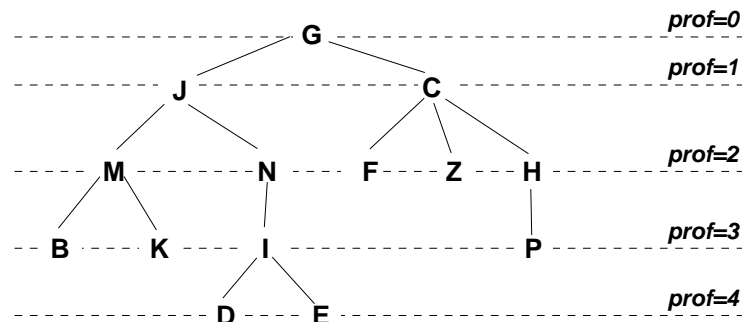
Examen Final. [11 de Diciembre de 2003]

Ej. 1.- [Primitivas (20 puntos)] Escribir las funciones del TAD PILA listadas a continuación, implementado por punteros o cursores ANULA(P), METE(x,P), SACAR(P), TOPE(P) y VACIA(P). Escribir todos los tipos, definiciones, funciones y procedimientos auxiliares necesarios.

Ej. 2.- [Ejercicios de programación (total 80 puntos)]

a) **[cuenta-prof (35 puntos)]** Escribir un función `function CUENTA_PROF(n:nodo; m:integer; A:arbol) : integer;` que dado un nodo `n` en un árbol `A` cuenta el número de nodos del subárbol de `A` cuya raíz es `n` que están a profundidad `m` o menor (con respecto a `n`). Por ejemplo, para el árbol de la figura debe retornar

CUENTA_PROF(G,2,A) -> 8
CUENTA_PROF(J,1,A) -> 3
CUENTA_PROF(N,3,A) -> 4



Usar las primitivas de árbol ordenado orientado siguientes:

HIJO_MAS_IZQ(n,A), HERMANO_DER(n,A). Hacer la función *recursiva*. Notar que, por ejemplo:

CUENTA_PROF(G,2,A) = 1 + CUENTA_PROF(J,1,A) + CUENTA_PROF(C,1,A)

La recursividad de la función debe cortar cuando $n = \Lambda$ o $m < 0$. El tiempo de ejecución del algoritmo debe ser $O(n)$, donde n es el número de nodos en el árbol.

b) **[semejante (10 puntos)]** Dos árboles ordenados T y T' se dicen “semejantes” si tienen la misma estructura. Formalmente esto significa que

- ambos son vacíos, ó
- ambos son no vacíos y los subárboles de sus hijos correspondientes son semejantes.

Informalmente decimos que T y T' tienen ambos la misma “forma”. **Consigna:** Escriba una función

`function SEMEJANTE(T,T': nodo) : boolean;`

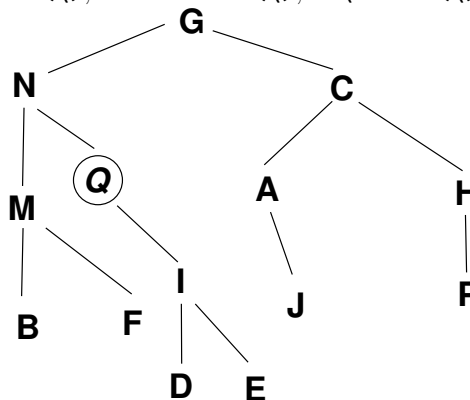
que retorna `true` si los árboles son semejantes y `false` caso contrario. Usar las primitivas de árbol ordenado orientado siguientes:

1) PADRE(n,A)

- 2) HIJO_MAS_IZQ(n, A)
 3) HERMANO_DER(n, A)
- c) **[junta2 (35 puntos)]** Escribir un procedimiento `procedure JUNTA2(var P: pila);` que modifica la pila P sumando de a dos los elementos en la pila. Si el número de elementos es impar, entonces el último queda inalterado. Por ejemplo, si $P = \{\text{tope} \rightarrow 1, 3, 5, 2, 4, 1, 2, 3, 9\}$ entonces después de hacer `JUNTA2(P)` debe quedar $P = \{\text{tope} \rightarrow 4, 7, 5, 5, 9\}$. Utilizar las funciones del TAD PILA `ANULA(P)`, `METE(x,P)`, `SACA(P)`, `TOPE(P)` y `VACIA(P)`.

Ej. 3.- [LIBRES. Ejercicios operativos (total 80 puntos)] Atención!! Alumnos libres deben completar un mínimo de 70 % en cada uno de los ítems

- a) **[reconstruir-arbol (20 puntos)]** Dibujar el árbol ordenado orientado cuyos nodos, listados en orden previo y posterior son
- $\text{ORD_PRE} = \{Z, A, B, J, D, E, K, L, M\}$,
 - $\text{ORD_POST} = \{A, D, K, M, L, E, J, B, Z\}$.
- b) **[abb (20 pto)]** Dados los enteros $\{1, 5, 8, 3, 2, 10, 20\}$ insertarlos, en ese orden, en un TAD ARBOL BINARIO DE BUSQUEDA. Mostrar las operaciones necesarias para eliminar los elementos 5, 6 y 3, en ese orden.
- c) **[huffman (20 puntos)]** Dados los caracteres siguientes con sus correspondientes probabilidades, contruir el código binario según el algoritmo de *Huffman* y encodar la palabra TRANSVERSAL $P(A) = P(N) = P(S) = P(L) = 0,2, P(R) = P(T) = P(V) = P(E) = 0,05$. Calcular la longitud promedio del código obtenido.
- d) **[particionar-arbol (10 puntos)]** Considerando el árbol de la figura, decir cuales son los nodos `DESCENDIENTES(Q)`, `ANTECESORES(Q)`, `IZQUIERDA(Q)` y `DERECHA(Q)`.



Ej. 4.- [LIBRES (20 pto, 5 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) Dadas las funciones

- ☐ ... $T_1(n) = \sqrt{n} + 0,5 n^{1,7} + 2,3 \log_2 n$
☐ ... $T_2(n) = 3 \cdot 2^n + 5 n^2 + 6 \sqrt{n}$
☐ ... $T_3(n) = 5n! + 2 \cdot 5^n + 4 n^5$
☐ ... $T_4(n) = 3\sqrt{n} + 2 n^2 + 4 n^3$

decir cuál de los siguientes ordenamientos es el correcto

Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]

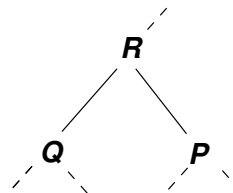
- ☐ ... $T_2 < T_1 < T_4 < T_3$
- ☐ ... $T_4 < T_1 < T_2 < T_3$
- ☐ ... $T_1 < T_4 < T_2 < T_3$
- ☐ ... $T_3 < T_4 < T_1 < T_2$

b) El mejor caso para el algoritmo de clasificación rápida (“quick-sort”) es cuando el pivote v particiona la secuencia en dos secuencias de longitud n_1 y n_2 tales que

- ☐ $n_1 \approx n_2$
- ☐ $n_1 \gg n_2$
- ☐ $n_1 \ll n_2$
- ☐ $n_1 \approx \sqrt{n_2}$

c) El montículo es un árbol binario que satisface la condición de ser “parcialmente ordenado”. Si R, P, Q son las etiquetas del nodo y sus dos hijos, la condición de parcialmente ordenado se expresa como (Nota: Consideramos un montículo “minimal”):

- ☐ ... $Q + P \leq R$.
- ☐ ... $R \leq Q, P$.
- ☐ ... $Q \leq R \leq P$.
- ☐ ... $P \leq R \leq Q$.



d) ¿Cuál es el tiempo de ejecución en promedio para inserción de un elemento una tabla de dispersión abierta con N elementos y B cubetas?

- ☐ ... $O(N/B)$
- ☐ ... $O(N)$
- ☐ ... $O(B)$
- ☐ ... $O(NB)$