

Algoritmos y Estructuras de Datos. Recuperatorio. [29 de Noviembre de 2007]

[Ej. 1] [clases (15 pts)]

- a) [parcial-1 lista (15 pts)] Escribir los siguientes métodos del **TAD lista**: `insert(p,x)`, `erase(p)`, `begin()`.
- b) [parcial-2 arbol-bin (15 pts)] Escribir los siguientes métodos del **TAD árbol binario**: `find(x,p)`, `insert(p,x)`, `clear()`.
- c) [parcial-3 set (15 pts)] Escribir los siguientes métodos del **TAD conjunto** por listas ordenadas: `insert(x)`, `find(x)`, `clear()`.

[Ej. 2] [programacion (50 pts)]

- a) [map-pre-post (15 pts)] Escribir una función
`void map_pre_post(tree<int> &T, list<int> &L, int (*fpre)(int), int (*fpost)(int))`
 que lista los valores nodales del árbol ordenado orientado T en una mezcla de orden previo y posterior, de acuerdo a la siguiente definición

$$\text{mpp}(\Lambda, f, g) = \text{lista vacía}$$

$$\text{mpp}(n, f, g) = f(n), \text{mpp}(n_1, f, g), \dots, \text{mpp}(n_m, f, g), g(n)$$

donde $n_1 \dots n_m$ son los hijos del nodo n . Por ejemplo, si $T = (1 \ 3 \ (5 \ 6 \ 7 \ 8))$, $f(x) = x$ y $g(x) = x + 1000$, entonces `map_pre_post(T, L, f, g)` debe dar
 $L = (1, 3, 1003, 5, 6, 1006, 7, 1007, 8, 1008, 1005, 1001)$.

- b) [maxdev (20 pts)]

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:

$\text{max_dev}(a_1, a_2, \dots, a_n) = (\max_{j=1}^n a_j) - (\min_{j=1}^n a_j)$. Escribir una función

`int max_dev_m(list<int> &l, int m)`; que retorna el máximo de las máximas desviaciones de las subsecuencias de L de longitud m , es decir

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

Por ejemplo, si $L = (1, 3, 5, 4, 3, 5)$, entonces `max_dev_m(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.

- c) [is-mapped-set (15 pts)] Escribir una función predicado
`bool is_mapped_set(set<int> &A, set<int> &B, int (*mapfun)(int))`; que retorna verdadero si el conjunto B contiene los elementos de A, mapeados vía la función `mapfun`. Por ejemplo, si $A = \{-5, -3, 5, 10\}$ y $B = \{9, 25, 100\}$ entonces `is_mapped_set(A, B, sq)`

Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]

debe retornar **true**, donde $\text{sq}(x)$ es la función que eleva al cuadrado, es decir $\text{sq}(x)=x*x$; . Mientras que si $A = \{-5, -3, 2, 10\}$ y $B = \{4, 9, 100, 130\}$, entonces debe retornar **false** ya que el $25 = (-5)^2$ no está en B y 130 no es el cuadrado de ningún elemento de A .
Sugerencia: Crear un conjunto temporario conteniendo las imágenes de los elementos de A y compararlo con B .

[Ej. 3] [operativos (25 pts)]

▪ [t-exec (5 pts)]

Dadas las funciones

- $T_1(n) = 5n^3 + 2n! + \log n$,
- $T_2(n) = 2^{15} + 2 \cdot 5^n + 3 \cdot n^2$,
- $T_3(n) = 6! + n^2 + n^{1.7}$,
- $T_4(n) = 1.3 \cdot 2^3 + 20n + \log_2 10$.

ordenarlas de menor a mayor.

$$T_{\square} < T_{\square} < T_{\square} < T_{\square}$$

- [huffman (5 pts)] Dados los caracteres siguientes con sus correspondientes probabilidades, contruir el código binario y encodar la palabra **CONEXA**

$P(C) = 0.2, P(O) = 0.1, P(N) = 0.1, P(A) = 0.05, P(L) = 0.05, P(X) = 0.1, P(E) = 0.1, P(Q) = 0.3$ Calcular la longitud promedio del código obtenido.

- [abb (10 pts)] Dados los enteros $\{7, 9, 12, 8, 4, 3, 1, 2, 5, 6\}$ insertarlos, en ese orden, en un “árbol binario de búsqueda”. Mostrar las operaciones necesarias para eliminar los elementos 4, 9 y 5, en ese orden.
- [hash-dict (5 pts)] Insertar los números 5, 18, 28, 11, 10, 38, 3, 2 en una tabla de dispersión cerrada con $B = 8$ cubetas, con función de dispersión $h(x) = x \% 8$ y estrategia de redistribución lineal.

[Ej. 4] [preguntas (10 pts, 2.5 por pregunta)]

a) ¿Cuál es el criterio para elegir una buena función de dispersión? ...

- ☐ Debe tratar de concentrar los elementos en pocas cubetas.
- ☐ Debe tratar de concentrar los elementos en una sólo cubeta.
- ☐ Debe tratar de concentrar los elementos en la primera cubeta.
- ☐ Debe distribuir los elementos en la forma más uniforme posible entre las cubetas.

b) Dado el árbol binario (**x e (d f g)**), ¿cuál de las siguientes opciones es verdadera?

- ☐ Es completo y es lleno.
- ☐ Es completo pero no lleno.
- ☐ Es lleno pero no completo.
- ☐ Ni es completo ni es lleno.

c) ¿Cuál es el número de niveles en un árbol binario lleno en función del número n de nodos en el árbol?

- ☐ ... $O(n \log n)$

Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]

- ☐ ... $O(1)$
- ☐ ... $O(n)$
- ☐ ... $O(\log n)$

d) Sea un tabla de dispersión abierta con B cubetas y n elementos. Asumiendo que la función de dispersión es lo suficientemente buena como para distribuir los elementos en forma uniforme entre las cubetas, el costo medio de inserción de un nuevo elemento es

- ☐ $O(n^2/B)$
 - ☐ $O((n/B)^2)$
 - ☐ $O(n + B)$
 - ☐ $O(1 + n/B)$
-