

Algoritmos y Estructuras de Datos.

2do Parcial. [27 de octubre de 2009]

ATENCIÓN: Para aprobar deben obtener un **puntaje mínimo** del 50 % en clases (Ej 1), 25 % en operativos (Ej 3) y un 60 % sobre las preguntas de teoría (Ej 4).

[Ej. 1] [clases (20pt)]

- a) **AOO:** declarar las clases `tree`, `cell`, `iterator`, (preferentemente respetando el anidamiento), incluyendo las declaraciones de datos miembros. Implementar el método `tree<T>::iterator tree<T>::insert(tree<T>::iterator n, const T& x)`
- b) **AB:** declarar las clases `btree`, `cell`, `iterator`, (preferentemente respetando el anidamiento), incluyendo las declaraciones de datos miembros. Implementar el método `btree<T>::iterator btree<T>::erase(btree<T>::iterator n);`

[Ej. 2] [programación (total 50pt)]

- a) **[split-down (25pt)]** Dados dos enteros M y $n > 0$, crear un AOO T tal que:
 - 1) La suma de los contenidos de las hojas es M , pero cada una de ellas es $h \leq n$.
 - 2) Se satisface que para cada nodo p la suma de sus hijos es $*p$.
 - 3) Cada nodo tiene a lo sumo g hijos, con $g > 1$ una constante dada.

Por ejemplo dados $M=20$, $n=4$, y $g=3$, un posible árbol (la solución no es única, por supuesto) es (20 (7 3 2 2) (7 3 2 2) (6 2 2 2)).

Consigna: Escribir una función `void split_down(tree<int> &T, int n, int g);` que realiza la tarea indicada.

Ayuda: El árbol se puede construir poniendo inicialmente M en la raíz, y dividiendo el contenido $*n$ de cada nodo en g valores aprox iguales hasta obtener valores $\leq n$. Para ello escribir una función recursiva auxiliar

`int depth_if(btree<int> &T, btree<int>::iterator n, pred_t pred);`. Asumir que existe una función `void distrib(int m, int g, vector<int> &v);` que distribuye el entero m en a lo sumo g partes, tratando de que difieran a lo sumo en una unidad, por ejemplo `distrib(20, 6, v);` hace $v=[4, 4, 3, 3, 3, 3]$.

- b) **[depth-if (25pt)]**
 Dado un AB T encontrar, la profundidad máxima de los nodos que satisfacen un cierto predicado (*profundidad condicionada*). Por ejemplo, si $T=(6 (7 9 (3 1)) 2)$, entonces `depth_if(T, even)` debe retornar 1, ya que el nodo par a mayor profundidad es 2, mientras que `depth_if(T, odd)` debe retornar 3, ya que el nodo impar a máxima profundidad es 1.
Consigna: Escribir una función `int depth_if(btree<int> &T, bool (*pred)(int));` que realiza la tarea indicada.
Ayuda: La función recursiva auxiliar debe retornar la máxima profundidad de un nodo que satisface el predicado o -1 si el árbol está vacío o ningún nodo satisface el predicado.
 Entonces, dadas las profundidades condicionadas d_l , d_r de los hijos la profundidad se puede expresar en forma recursiva como

$$\text{depth}(n) = \begin{cases} -1, & \text{si } n \text{ es } \Lambda, \\ \max(d_r, d_l) + 1, & \text{si } \max(d_r, d_l) \geq 0, \\ 0, & \text{si } n \text{ satisface el predicado,} \\ -1, & \text{si } n \text{ no satisface el predicado.} \end{cases}$$

Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]

[Ej. 3] [operativos (total 20pt)]

- a) [rec-arbol (10pt)] Dibujar el AOO cuyos nodos, listados en orden previo y posterior son
- $ORD_PRE = \{W, X, Y, A, B, C, D, F, G, H\};$
 - $ORD_POST = \{X, Y, B, F, G, H, D, C, A, W\}.$
- b) [huffman (10pt)] Dados los caracteres siguientes con sus correspondientes probabilidades, contruir el código binario utilizando el algoritmo de Hufmann y encodar la palabra SUDAFRICA $P(S) = 0.15, P(U) = 0.15, P(D) = 0.05, P(A) = 0.05, P(F) = 0.20, P(R) = 0.10, P(I) = 0.20, P(C) = 0.10$. Calcular la longitud promedio del código obtenido. Justificar si cumple o no la condición de prefijos.

[Ej. 4] [Preguntas (total = 20pt, 4pt por pregunta)]

- a) Defina que es un **camino** en un árbol. Dado el AOO (a b (c e (f g)) d)), cuáles de los siguientes son caminos?
- 1) (a, c, f, g),
 - 2) (e, c, f, g),
 - 3) (e, c, a),
 - 4) (c, f, g).
- b) Defina qué es un **AB lleno**. ¿Cuál es el **número de nodos** en el nivel l de un árbol lleno? ¿Cuál es el **número total de nodos** en un árbol lleno de n niveles? ¿Cuántas **hojas** contiene?
- c) ¿Es verdad que si dos nodos están en el **mismo nivel** de un árbol, entonces son **hermanos**? ¿Y la recíproca? De ejemplos.
- d) ¿Es posible **insertar** en una posición **no-dereferenciable** (Λ) en un AB? ¿Y en un AOO? Discuta y de ejemplos.
- e) ¿Porqué el AB correspondiente a la **codificación binaria** de una serie de caracteres debe ser un **ABC (AB completo)**?