

Algoritmos y Estructuras de Datos. 2do Parcial. [2012-10-25]

ATENCIÓN (1): Para aprobar deben obtener un **puntaje mínimo** de

- 50 % en clases (Ej 1),
- 50 % en programación (Ej 2),
- 50 % en operativos (Ej 3) y
- 60 % sobre las preguntas de teoría (Ej 4).

ATENCIÓN (2): Recordar que tanto en las clases (Ej. 1) como en los ejercicios de programación (Ej 2.) **deben usar la interfaz STL.**

[Ej. 1] [clases (20pt)] Insistimos: deben usar la interfaz STL (“AVANZADA”).

- a) **[AOO (10pt)]** Para el TAD Arbol Ordenado Orientado (AOO):
- Declarar las clases `tree`, `cell`, `iterator`, con su correspondiente anidamiento, incluyendo las declaraciones de datos miembros.
 - Implementar los métodos
 - `tree<T>::iterator tree<T>::insert(tree<T>::iterator n, const T& x);`
 - `tree<T>::iterator tree<T>::begin();`
- b) **[AB (10pt)]** Para el TAD Arbol Binario (AB):
- Declarar las clases `btree`, `cell`, `iterator`, con su correspondiente anidamiento, incluyendo las declaraciones de datos miembros.
 - Implementar los métodos
 - `btree<T>::iterator btree<T>::erase(btree<T>::iterator n);,`
 - `btree<T>::iterator btree<T>::iterator::right();.`

[Ej. 2] [Programación (total = 40pt)] Insistimos: deben usar la interfaz STL.

Atención: Hay 3 ejercicios cuya suma es **60**, pero el total de la sección es **40**. (O sea, la nota final en esta sección es $\min(40, n1+n2+n3)$).

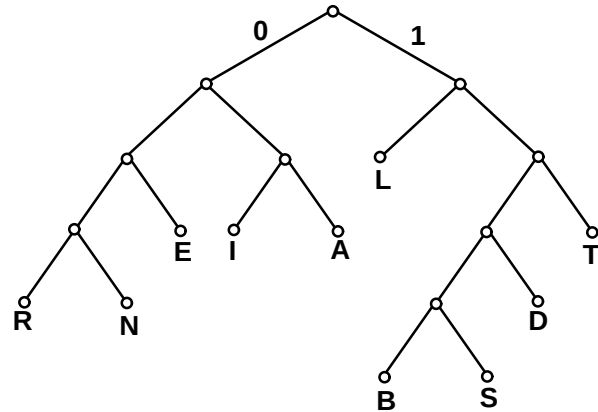
- a) **[bin2ord (20pt)]** Escribir una funcion `void bin2ord(btree<int> &B, tree<int> &T, int LAMBDA);` que dado un árbol binario (AB) B de enteros **positivos** lo convierte a un árbol ordenado orientado (AOO) T con la siguiente convención. En caso de que uno de los nodos del AB tenga uno solo de los hijos reemplazamos el valor por un valor especial LAMBDA (en este caso puede ser `LAMBDA=-1`, ya que todos los valores nodales son >0). Por ejemplo, si `B=(1 (2 . (3 . 4)) .)`, y `LAMBDA=-1` entonces debe dar `T=(1 (2 -1 (3 -1 4)) -1)`. Notar que en notación Lisp esto equivale a reemplazar los puntos por LAMBDA.
- b) **[ord2bin (20pt)]** Escribir la funcion inversa `void ord2bin(tree<int> &T, btree<int> &B, int LAMBDA);` que dado un AOO (se asume que fue generado por `bin2ord`) lo convierte de nuevo a AB. (Debería ser siempre `B==ord2bin(bin2ord(B))`).
- c) **[lesser (20pt)]** Se define una **relación de orden** entre AOO de enteros de la siguiente forma: `A<B` si `(na,c0a,c1a,c2a...)<(nb,c0b,c1b,c2b...)`, donde `na` es la raíz de A, `h0a` el **subárbol** del primer hijo de A y así siguiendo. En la expresión anterior se toma el orden lexicográfico para listas y se asume que en caso de tener longitudes diferentes se completa con $-\infty$.
Consigna: Escribir un predicado binario `bool lesser(tree<int> &A, tree<int> &B);` que determina si `A<B` en el sentido indicado previamente.
Nota: puede demostrarse que efectivamente este predicado binario es una **relación de orden total**.

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

- a) **[rec-arbol (8pt)]** Dibujar el AOO cuyos nodos, listados en orden previo y posterior son

- $ORD_PRE = \{W, X, Y, Z, T, R, S, Q\}$,
 - $ORD_POST = \{X, Z, T, R, Q, S, Y, W\}$,
- b) [huffman (6pt)] Dados los caracteres siguientes con sus correspondientes probabilidades, contruir el código binario utilizando el algoritmo de Hufmann y encodar la palabra FRAGATA
 $P(G) = 0.10, P(F) = 0.10, P(R) = 0.10, P(T) = 0.30, P(A) = 0.05, P(S) = 0.35$. Calcular la longitud promedio del código obtenido.
- c) [hf-decode (6pt)]

Utilizando el código de la derecha desencodar el mensaje 100101100000100001110111101



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

- a) Dado el árbol ordenado orientado (AOO) (W X Y (A B (C (D F G H)))) determinar cuales son los nodos **antecesores**, **descendientes**, **izquierda** y **derecha** del nodo A. ¿Son **disjuntos**? Justifique.
- b) Si queremos generar un código binario de igual longitud para un conjunto de 26 caracteres. ¿Cuántos bits tendrá, como **mínimo**, cada caracter?
- c) ¿Como se define la **altura** de un nodo en un árbol? ¿Cuál es la altura de los nodos A, B, y C en (W X Y (A B (C (D F G H))))?
- d) ¿Cómo se define la clase **iterator** para árboles binarios? Enumere sus campos dato y discuta la utilidad de cada uno de ellos. De ejemplos.
- e) ¿Cómo se define en forma recursiva el **orden posterior** para un árbol ordenado orientado (AOO)?