

Algoritmos y Estructuras de Datos. 2do Parcial. Tema: 1A. [27 de Mayo de 2004]

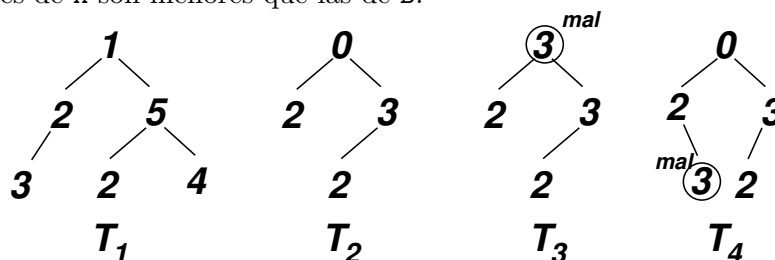
[Ej. 1] [Clases (20 puntos)] Escribir la implementación en C++ del TAD `Arbol binario` (clase `btree`) implementado con celdas enlazadas por punteros. Las funciones a implementar son `insert(key, val)`, `retrieve(key)`, `erase(p)`, `begin()`, `end()`, `clear()` además de las clases `iterator` y `cell`.
Observaciones:

- Incluir las definiciones de tipo (`typedef`) y clases auxiliares necesarias.
- Se puede escribir la interface avanzada (con templates, clases anidadas, sobrecarga de operadores).

[Ej. 2] [Programación (total = 60 puntos)]

a) **[contenido (30 puntos)]**

Escribir una función predicado `bool contenido(btree<int> &A, btree<int> &B)`; que retorna verdadero si la estructura del árbol binario A esta “contenido” dentro de la de B y las etiquetas correspondientes de A son menores que las de B.



Para los árboles de la figura debemos tener .

- `contenido(T2,T1)` retorna `true`
- `contenido(T3,T1)` retorna `false`
- `contenido(T3,T1)` retorna `false`

En los casos que debe retornar false, el nodo que viola la condición está marcado. Se sugiere escribir una función auxiliar recursiva.

- b) **[nodos-n (15 puntos)]** Escribir una función `int nodos_n(tree<int> &A, int n)`; que cuenta el número de nodos de un árbol ordenado orientado cuya etiqueta es exactamente igual a n.
- c) **[verifica-par (15 puntos)]** Escribir una función predicado `bool verifica_par(tree<int> &A)`; que verifica si las etiquetas de todos los nodos de un árbol ordenado orientado A son pares.

[Ej. 3] [operativos (total = 10 puntos)]

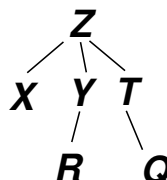
- **[rec-arbol (10 ptos)]** Dibujar el árbol ordenado orientado cuyos nodos, listados en orden previo y posterior son
 - `ORD_PRE` = {Z, A, X, U, V, Q, W, B, R, T},
 - `ORD_POST` = {X, U, Q, V, W, A, T, R, B, Z}.
- **[arbol-expr (10 ptos)]** Escribir el árbol que corresponde a la expresión $a * (b + c) / (d + z) - x * (y - w)$

Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]

[Ej. 4] [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!!**]



- ☐ ... Z
- ☐ ... X
- ☐ ... Y
- ☐ ... R

¿Cuál es el nodo que está a la izquierda de Q y no es antecesor de R

Sea el árbol ordenado orientado $A=(4 \ 7 \ 6 \ (2 \ 3))$. Si hacemos las operaciones

```
n = A.find(7);  
n++; n++;  
n = A.insert(n,1);
```

¿Cuál de las siguientes afirmaciones es verdadera?

- ☐ ... Da un error
- ☐ ... Queda $A=(4 \ 7 \ 6 \ (2 \ 1 \ 3))$
- ☐ ... Queda $A=(4 \ 7 \ 6 \ (2 \ 3) \ 1)$
- ☐ ... Queda $A=(4 \ 7 \ 6 \ 1 \ (2 \ 3))$

Si $A=(3 \ 5 \ 4 \ (6 \ 7 \ 8))$ y $B=(5 \ 3)$ entonces

¿cómo quedan A y B después de hacer...?

```
n=A.find(6);  
m=B.begin(); m=m.lchild();  
m=m.lchild();  
B.splice(m,n);
```

- ☐ ... Da un error.
- ☐ ... $A=(3 \ 5 \ 4)$ y $B=(5 \ (3 \ (6 \ 7 \ 8)))$
- ☐ ... $A=(3 \ 5 \ 4 \ 6)$ y $B=(5 \ (3 \ 7 \ 8))$
- ☐ ... $A=(3 \ 5 \ 4)$ y $B=(5 \ (3 \ (6 \ 7 \ 8)))$

¿Cuál de los siguientes árboles binarios es “completo”?

- ☐ ... (1 2 (3 . 7))
- ☐ ... (1 2 (3 5 7))
- ☐ ... (1 2 (3 5 .))
- ☐ ... (1 (2 . 4) (3 5 7))