

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

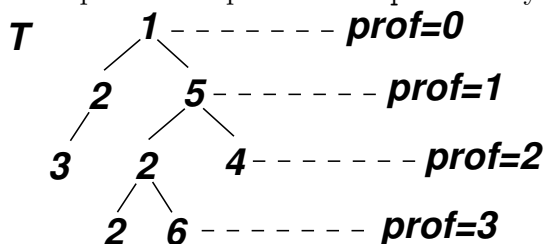
[Ej. 1] [Clases (20 puntos)] Escribir la implementación en C++ del TAD Arbol Ordenado Orientado (clase `tree`) implementado con celdas enlazadas por punteros. Las funciones a implementar son `insert(key, val)`, `retrieve(key)`, `erase(p)`, `begin()`, `end()`, `clear()` además de las clases `iterator` 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) [cant-nodos-prof (30 puntos)]

Escribir una función `int cant_nodos_prof(btrees<int> &A, int prof)`; que retorna el número de nodos de un árbol binario `A` que están a profundidad `prof` o mayor.



Para el árbol  $T$  de la figura debemos tener

- `cant_nodos_prof(T,0)` retorna 8
- `cant_nodos_prof(T,1)` retorna 7
- `cant_nodos_prof(T,2)` retorna 5
- `cant_nodos_prof(T,3)` retorna 2
- `cant_nodos_prof(T,4)` retorna 0
- `cant_nodos_prof(T,5)` retorna 0

Se sugiere escribir una función auxiliar recursiva.

- b) [algun-par (15 puntos)] Escribir una función predicado `bool algun_par(trees<int> &A)`; que verifica si *al menos una* de las etiquetas de un árbol ordenado orientado `A` es par.
- c) [nodos-mayores-que (15 puntos)] Escribir una función `int nodos_mayores_que(trees<int> &A, int m)`; que cuenta el número de nodos de un árbol ordenado orientado cuya etiqueta es mayor o igual que `m`.

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

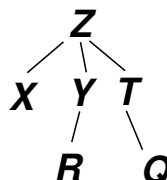
- [rec-arbol (10 ptos)] Dibujar el árbol ordenado orientado cuyos nodos, listados en orden previo y posterior son
  - `ORD_PRE` = { $W, A, B, D, E, H, J, C, F, G$ },
  - `ORD_POST` = { $D, J, H, E, B, F, G, C, A, W$ }.
- [arbol-expr (10 ptos)] Escribir el árbol que corresponde a la expresión  $(a - q) * (r + z) / (w - f) + d * e / (x - y)$

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!!**]



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

¿Cuál es el nodo que está a la derecha de X y es antecesor propio de Q?

Sea el árbol ordenado orientado A=(5 3 (4 7 6) 2). Si hacemos las operaciones

n = A.find(3);

n++;

n = A.insert(n,5);

¿Cuál de las siguientes afirmaciones es verdadera?

- ☐ ... Da un error
- ☐ ... Queda A=(5 3 (4 5 7 6) 2)
- ☐ ... Queda A=(5 3 5 (4 7 6) 2)
- ☐ ... Queda A=(5 3 (4 7 6) 5 2)

Si A=(2 (3 (6 4 2)) 5) y B=(1 7 9) entonces ¿cómo quedan A y B después de hacer...?

n=A.find(6);

m=B.begin(); m=m.lchild();

m++;

m=m.lchild();

B.splice(m,n);

- ☐ ... A=(2 (3 (6 4 2)) 5) y B=(1 7 (9 (6 4 2)))
- ☐ ... A=(2 3 (5 6)) y B=(1 7 (9 4 2))
- ☐ ... A=(2 3 5) y B=(1 7 (9 (6 4 2)))
- ☐ ... Da un error.

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

- ☐ ... (5 (6 7 8) .)
- ☐ ... (5 (6 7 8) 3)
- ☐ ... (5 (6 7 8) (3 . 1))
- ☐ ... (5 (6 7 8) (3 4 .))