

Algoritmos y Estructuras de Datos.

Parcial 1. [2022-10-06]

- [ATENCIÓN 1]** Para aprobar deben obtener un **puntaje mínimo** del 60 % en las preguntas de teoría y 50 % en clases y operativos.
- [ATENCIÓN 2]** Escribir cada ejercicio en **hoja(s) separada(s)**. Es decir todo **CLAS2** en una o más hojas **separadas**, **OPER2** en una o más hojas **separadas**, **PREG2** en una o más hojas **separadas**, etc...
- [ATENCIÓN 3]** Encabezar las hojas con **sección**, **Nro de hoja (relativo a la sección)**, **apellido**, y **nombre**, ASI:

CLAS2, Hoja #2/3	TORRES, LINUS
------------------	---------------

[Ej. 1] [CLAS1 (W=20pt)]

a) [list].

- Indique un prototipo (definición de atributos y métodos públicos y privados) de la clase lista implementada como celdas simplemente enlazadas.
- Implemente los métodos **begin** y **end** (deben ser $O(1)$)
- Implemente el método **erase(iterator_t p, iterator_t q)** que elimina el rango entre los elementos indicados por **p** y **q** respectivamente.

b) [stack].

- Indique un prototipo de una implementación del TAD Pila
- Implemente los métodos
 - **T max(stack::<T> &S)**; que devuelva el máximo valor almacenado en la pila
 - **void erase(stack::<T> &S, T t)**, que elimina de la pila los objetos **t**.

c) [A00].

- Indique un prototipo de implementación de un árbol binario.
- Implemente un iterador adecuado para dicho prototipo de TAD.

[Ej. 2] [OPER1 (W=20pt)]

- a) **[rec-arbol]**: Dibujar el A00 cuyos nodos, listados en orden previo y posterior son

```
1   ORD-PRE = (Z W A B C D X Y T)
2   ORD-POST = (W B A X T Y D C Z)
```

luego particionar el conjunto de nodos respecto al nodo A.

- b) **[operaciones]**: Sea el árbol $D = (6 \ (5 \ 4 \ 3) \ (2 \ 7 \ 9 \ 8) \ (1 \ 0))$. Después de hacer:

```
1 auto n = D.find(5);
2 n++;
3 n = n.lchild();
4 n++;
5 n = n.lchild();
6 n = D.insert(n, 2);
```

¿Cómo queda el árbol? ¿Se produce un error?

- c) **[Notación $O(\cdot)$]**. cada una de las funciones T_1, \dots, T_4 determinar su velocidad de crecimiento (expresarlo con la notación $O(\cdot)$) y ordenarlas de forma creciente.

$$T_1 = 2n^2 + 2\log_{10} n + 4n^4 + \sqrt[2]{n}$$

$$T_2 = 4n^3 + 3n! + 2 \cdot 3^n$$

$$T_3 = 2n^4 + 3 \cdot 3^n + 3^4$$

$$T_4 = \log_2 n + 4$$

$$T_5 = \log(8)n^{1.5} + 3.2\log n + 4^4$$

- d) [hacer-arbol (5pt)] Utilizando sólo métodos `insert`, `lchild`, `n++`, e iteradores del Arbol Ordenado Orientado AOO, complete el siguiente código que arma el árbol $T=(3 \ 5 \ 6 \ 8) \ 4)$

```
1    tree<int> T;  
2    tree<int>::iterator n = T.insert(T.begin(),3);  
3    ...  
4    COMPLETAR  
5    ...
```

[Ej. 3] [PREG1 (W=20pt)]

- a) Se puede insertar un elemento en una posición **dereferenciable** en una lista? Y en una **no-dereferenciable**?
- b) Enuncie y justifique la **regla de transitividad** para la notación $O(\cdot)$. De un ejemplo.
- c) En una **correspondencia**: ¿puede haber una **misma clave** con **diferentes valores**? Por ej. $M=(5 \rightarrow 4, \ 5 \rightarrow 2)$ ¿Puede haber **diferentes claves** con el **mismo valor**? Por ej. $M=(3 \rightarrow 5, \ 4 \rightarrow 5)$
- d) En un árbol, ¿puede un nodo tener **más de un padre**? ¿Puede un nodo **no tener padre**?
- e) Sea el árbol $(z \ r \ (p \ (q \ n) \ (t \ m)))$.
- Cuáles de los siguientes son **caminos**?
 - $(p \ t)$
 - $(n \ q \ p)$
 - $(t \ p \ q)$
 - $(z \ p \ q)$
 - Liste las **hojas** del árbol y los nodos a **profundidad 2**.
 - ¿Cuál es la **altura** del árbol?
- f) Defina que entiende por un **algoritmo de búsqueda exhaustiva** y un **algoritmo ávido**. Explique como se aplicarían estas estrategias a un problema como el de coloración de grafos o el del agente viajero (TSP).
- g) Si se implementa una **pila** en términos de **listas simplemente enlazadas**: ¿Dónde ubicaría el **tope** de la pila? ¿En el comienzo o en el fin de la lista? ¿Porqué?