

## Algoritmos y Estructuras de Datos. Parcial 1. [2017-10-05]

1. **[ATENCIÓN 1]** Para aprobar deben obtener un **puntaje mínimo** del 60 % en las preguntas de teoría y 50 % en clases y operativos.
2. **[ATENCIÓN 2]** Escribir cada ejercicio en **hoja(s) separada(s)**. Es decir todo **CLAS1** en una o más hojas **separadas**, **OPER1** en una o más hojas **separadas**, **PREG1** en una o más hojas **separadas**, etc...
3. **[ATENCIÓN 3]** Encabezar las hojas con **sección, Nro de hoja (relativo a la sección), apellido, y nombre, ASI:**

|                  |                 |
|------------------|-----------------|
| CLAS1, Hoja #2/3 | TORVALDS, LINUS |
|------------------|-----------------|

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

Recordar que **deben usar la interfaz STL**.

- a) Dada la siguiente implementación de celdas para una lista simplemente enlazada:

```

1      template<class T> class cell{
2          friend class list;
3          T elem;
4          cell* next;
5          cell() : next(NULL) {}
6      };
7      typedef cell *iterator;
```

Implementar los métodos:

```

1      iterator insert(iterator p, T elem);
2      iterator erase(iterator p);
3      iterator erase(iterator p, iterator q);
```

**Nota:** La implementación debe ser tal que las dos primeras (**insert(p,x)** y **erase(p)**) deben ser  $O(1)$ .

- b) En una implementación de Árbol Ordenado Orientado AOO en donde cada nodo del árbol es representado por una estructura como la siguiente:

```

1      template<class T> struct cell{
2          cell * left_child, right;
3          T elem;
4      }
```

- ¿Por qué es insuficiente utilizar iteradores que sean **typedef cell\* iterator**? Presente un ejemplo que lo demuestre.
- Proponga una definición de iterador que solucione el/los problemas mencionados en el punto anterior.

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

- a) **[rec-arbol (5pt)]** Dibujar el Árbol Ordenado Orientado AOO cuyos nodos, listados en orden previo y posterior son

- ORD-PRE=(W,A,B,C,T,U,M,N),
- ORD-POST=(A,T,M,N,U,C,B,W),

b) Dada la siguiente función:

```

1      map<int,string>::iterator
2          f1(map<int,string>&M, list<int>&L){
3          auto it = L.begin();
4          while(it!=L.end()){
5              auto m = M.find(*it);
6              if(m!=M.end())
7                  return m;
8              it++;
9          }
10         return M.end();
11     }
```

Determine el tiempo de ejecución  $T(n)$  para el peor caso, mejor caso y el caso promedio.

**Nota:** Asumir que la correspondencia está implementada con **vectores ordenados**.

c) **[part-arbol (5pt)]** Dado el Árbol Ordenado Orientado (AOO):

(A B C (D F (G H)) (E I))

determinar cuales son los nodos **antecedentes propios**, **descendientes propios**, **izquierda y derecha** del nodo D. ¿Son **disjuntos**? Justifique.

d) **[hacer-arbol (5pt)]** Utilizando sólo métodos **insert**, **lchild**, **right** e iteradores del Arbol Ordenado Orientado AOO, complete el siguiente código que arma el árbol  $T=(7 \ 9 \ (8 \ 3))$

```

1      tree<int> T;
2      tree<int>::iterator n = T.insert(T.begin(),7);
3      ...
4      COMPLETAR
5      ...
```

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

a) Notación  $O(\cdot)$ .

- Para cada una de las funciones  $T_1, \dots, T_5$  determinar su velocidad de crecimiento (expresarlo con la notación  $O(\cdot)$ ).

$$T_1 = 6n^4 + 3\log_{10} n + \sqrt[4]{n} + 4n^3,$$

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

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

$$T_4 = \log_{10} n + 3,$$

$$T_5 = \log(15)n^{1.5} + 3^2 + 2.4 \log n.$$

- Ordenar las funciones por tiempo de ejecución, es decir ponerlas en orden de la siguiente forma:  $T_x < T_y < T_z \dots$

b) ¿Cuál es el tiempo de ejecución (mejor/promedio/peor) para el algoritmo de **búsqueda binaria** en un vector ordenado?

c) Considerando la implementación de pila con listas simplemente enlazadas, ¿Cuál es el tiempo de ejecución (mejor/promedio/peor) de **pop**, **push** y **top** cuando el tope de la pila está en el comienzo de la lista? ¿Y cuando el tope está al final?

d) Sea el árbol (z (q p r) (d (e a b))). Cuáles de los siguientes son **camino**s?

- (z d e)
- (e d z)
- (q r)

■ (a e b)

- e) ¿Existe una relación biunívoca entre un Arbol Ordenado Orientado (AOO) y su **orden previo**? ¿Y con el **orden posterior**? ¿Y con ambos a la vez? De ejemplos.
- f) Explique qué se entiende por algoritmos de **búsqueda exhaustiva** y **heurístico**. Discuta cuales son las ventajas y desventajas.
- g) Exprese la **regla del producto** para la notación  $O(.)$ . De un ejemplo.
- h) Dado el siguiente AOO (15 (2 (7 11 20 (24 1))) (8 9))
- ¿Donde queda apuntado **n** después de las siguientes operaciones?
- ```
1      n = T.find(2);
2      n = n.lchild();
3      n = n.lchild();
4      n++;n++;
5      n = n.lchild();
```
- ¿Como queda el árbol al hacer **T.erase(T.find(7));** ?
- i) Comente ventajas y desventajas del uso de listas **doblemente enlazadas** con respecto a **simplemente enlazadas**.
- j) Discuta si es posible insertar en una **posición dereferenciable** en Arbol Ordenado Orientado (AOO). ¿Y en una **no dereferenciable**?