Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]

Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática
Algoritmos y Estructuras de Datos

Algoritmos y Estructuras de Datos.
Parcial 1. [2016-10-06]

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 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 LOVELACE, ADA`

[Ej. 1] [CLAS1 (W=20pt)]
Recordar que **deben usar la interfaz STL**.

a) **[lista]** Implemente los métodos:

```
1       iterator_t insert(T& x);
2       iterator_t erase(iterator_t p);
3       iterator_t erase(iterator_t p, iterator_t q);
```

de una lista doblemente enlazada por punteros, siendo:

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

b) Para garantizar $T(n) = O(\log(n))$ en las búsquedas en un mapa dada una clave es necesario mantener ordenadas las mismas. Utilizando la implementación de mapa basada en vectores, implemente el método:

```
1       iterator insert(T_k key, T_v value)
```

Para ello utilice e implemente el método

```
1       iterator lower_bound(T_k key)
```

que retorna un iterador a:

- Si existe la clave, la posición en la que ya se encuentra.
- Si no existe la clave, la primera posición en la cual puede insertarse la nueva entrada, preservando el orden de las claves.

Considere: `typedef int iterator`

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

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

- `ORD-PRE=(Z,Q,R,T,W,A,X,F,Y,U)`,
- `ORD-POST=(Q,R,A,W,F,X,U,Y,T,Z)`.

b) **[part-arbol (XXpt)]** Dado el árbol ordenado orientado (AOO):
`(Z Q (R (T U (V X) (W Y))))`
determinar cuales son los nodos **antecesores propios, descendientes propios, izquierda** y **derecha** del nodo `V`. ¿Son **disjuntos**? Justifique.

c) Un zoológico quiere construir hábitats naturales en los que exhibir los animales que dispone. Por desgracia, algunos animales se comerían a otros si se les diese la oportunidad. ¿Cómo utilizar un grafo para modelar esta situación? ¿Cómo utilizar una coloración del grafo para determinar el número de hábitats diferentes que se necesitan y la distribución de los animales en esos hábitats?

d) Utilizando sólo métodos **`insert, lchild, right`** e iteradores del AOO, complete el siguiente código que arma el árbol **`T=(3 (2 7 5) 9)`**

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

e) Considere el siguiente algoritmo:

```
1       bool a(int n) {
2         if(n==0) return 0;
3         if(n==1) return 1;
4         return a(n-1)+2*a(n-2);
5       }
```

Halle la expresión recursiva de los términos de la sucesión que se implementa. Determine la complejidad algorítmica en notación $O(\cdot)$

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

a) Ordenar las siguientes funciones por tiempo de ejecución. Además, para cada una de las funciones $T_1, \ldots, T_5$ determinar su velocidad de crecimiento (expresarlo con la notación $O(\cdot)$).

$$T_1 = 5\log_2 n + \sqrt[3]{n} + 2\,n^3 + 5\,n^2$$
$$T_2 = 2n^2 + 2 \cdot 3^n + 7^4$$
$$T_3 = 3 \cdot 3^n + 4n^5 + 6\,n!$$
$$T_4 = \log_{10} n + 3$$
$$T_5 = 5^6 + 3.4\log_2 n + \log_4(16)n^{2.5}$$

b) Se puede insertar un elemento en una posición **dereferenciable** en una lista? Y en una **no-dereferenciable**?

c) Enuncie y justifique la **regla de la suma** para la notación $O(\cdot)$. De un ejemplo.

d) En una **correspondencia**: ¿puede haber una **misma clave** con **diferentes valores**? Por ej. **`M=(1->2, 1->3)`** ¿Puede haber **diferentes claves** con el **mismo valor**? Por ej. **`M=(1->2, 3->2)`**

e) En un árbol, ¿puede un nodo tener **más de un padre**? ¿Puede un nodo **no tener padre**?

f) En un Arbol Ordenado Orientado (AOO): ¿importa el **orden de los nodos**? Por ejemplo, ¿es lo mismo el árbol **`(a b c)`** que el árbol **`(a c b)`**?

g) ¿Cuantas posiciones **no dereferenciables** puede haber en una **lista**? ¿Y en un **árbol**?

h) Dado dos nodos de un Arbol Ordenado Orientado (AOO) **`a,b`**, ¿Puede ser **b descendiente** y **antecesor** de **a** al mismo tiempo? ¿Puede ser **b descendiente propio** y **antecesor propio** de **a** al mismo tiempo?

i) Supongamos que tenemos un árbol (AOO) **`T=(1 3 (5 6 8) 7)`** y un iterator **`p`** apuntando al 5. ¿Cómo queda **`T`** si hacemos una inserción: **`T.insert(p,4);`**

j) Supongamos que tenemos un árbol (AOO) **`T=(2 (4 6 8) (7 10))`** y un iterator **`p`** apuntando al 4. ¿Cómo queda **`T`** si hacemos una supresión: **`T.erase(p);`**

mstorti@galileo/aed-3.0-36-g76cec07/Wed Sep 28 14:15:44 2016 -0300