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. [2015-09-17]

ATENCIÓN 1: Para aprobar deben obtener un **puntaje mínimo** del 60 % en las preguntas de teoría (Ej 2) y 50 % en clases.
ATENCIÓN 2: Hacer cada ejercicio en una **hoja separada**.

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

a) **[list]**
Siendo la siguiente una posible implementación del TAD Lista mediante celdas simplemente enlazadas por punteros:

```
1  class cell;
2  typedef cell *iterator_t;
3
4  class list {
5  private:
6    cell *first;
7    int _size;
8  public:
9    list();
10   ~list();
11   iterator_t insert(iterator_t p,elem_t j);
12   iterator_t erase(iterator_t p);
13   iterator_t erase(iterator_t p,iterator_t q);
14   void clear();
15   iterator_t begin();
16   iterator_t end();
17   iterator_t next(iterator_t p);
18   iterator_t prev(iterator_t p);
19   elem_t & retrieve(iterator_t p);
20   int size();
21 };
22
23 class cell {
24   friend class list;
25   elem_t elem;
26   cell *next;
27   cell() : next(NULL) {}
28 };
```

1) Implemente el método **`iterator_t insert(iterator_t p,elem_t j);`** que inserta en la lista un elemento **`j`** en la posición **`p`** (debe ser $O(1)$)
2) Implemente el método **`iterator_t erase(iterator_t p);`** que elimina de la lista el elemento ubicado en la posición **`p`** (debe ser $O(1)$)
3) Implemente el método **`int size();`** que retorna el número de elementos de la lista (debe ser $O(1)$)

b) **[map]** Siendo la siguiente una posible implementación del TAD Correspondencia mediante vectores ordenados:

```
1  class elem_t {
2  private:
3    friend class map;
4    domain_t first;
5    range_t second;
6  };
7
8  typedef elem_t* iterator_t;
9
10 class map {
11 private:
12   vector<elem_t> v;
13   iterator_t lower_bound(domain_t key);
```

```
14 public:
15   map():
16     iterator_t find(domain_t key);
17   iterator_t insert(domain_t key,range_t val);
18   range_t &retrieve(domain_t key);
19   bool empty();
20   void erase(iterator_t p);
21   int erase(domain_t key);
22   iterator_t begin();
23   iterator_t end();
24   void clear();
25   int size();
26   domain_t key(iterator_t p);
27   range_t &value(iterator_t p);
28 };
```

1) Implemente los métodos
   - **`iterator_t lower_bound(domain_t key);`** es la función auxiliar que retorna la primera posición donde el elemento podría ser insertado de manera que se mantenga ordenado.
   - **`iterator_t find(domain_t key);`**

   La implementación de ambos debe ser tal que **`find()`** es $O(\log(n))$

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

a) Considere el siguiente algoritmo. Comente brevemente qué es lo que resuelve el mismo.

```
1    bool algoritmo(list<int> &L, list<int>::iterator p, int x) {
2       if(p==L.end()) return false;
3       if(*p==x) return true;
4       return algoritmo(L,++p,x);
5    }
```

b) Determine el tiempo de ejecución $T(n)$ del algoritmo previo para:
1) El peor caso $T_{\text{peor}}(n)$
2) El mejor caso $T_{\text{mejor}}(n)$
3) El caso promedio $T_{\text{prom}}(n)$

c) ¿De qué complejidad algorítmica $O(n)$ es?

d) 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 $O(\,\cdot\,)$).

$$
\begin{aligned}
T_1 &= n^2 + 2 \cdot 4^n + 5^3 \\
T_2 &= 3 \cdot 2^n + 5n^4 + 2\,n! \\
T_3 &= 2 \log_5 n + \sqrt{n} + 3\,n^2 + 2\,n^5 \\
T_4 &= 4^5 + 2.7 \log_4 n + \log_2(5) n^{1.5} \\
T_5 &= \log_2 n + 5
\end{aligned}
$$

e) ¿Cuáles son los tiempos de ejecución para los diferentes métodos de la clase **`map<>`** implementada con **listas ordenadas** y **vectores ordenados** en el caso promedio? Métodos: **`find(key)`**, **`M[key]`**, **`erase(key)`**, **`erase(p)`**, **`begin()`**, **`end()`**, **`clear()`**.

f) Discuta las ventajas y desventajas de utilizar listas **doblemente enlazadas** con respecto a las **simplemente enlazadas**. ¿Cuáles son los métodos cuyo **tiempo de ejecución cambia** y por qué?

mstorti@galileo/parc1-2015-8-g5ff6a2a/Wed Sep 23 14:30:34 2015 -0300