

# CC3001 Algoritmos y Estructuras de Datos

## Control 1

Profesores: Patricio Poblete y Nelson Baloian

7 de abril de 2011

1.

- a) En clases se vio un algoritmo de multiplicación de polinomios que dividía cada polinomio en dos mitades:

$$A(x) = A_1(x)x^{n/2} + A_0(x)$$

$$B(x) = B_1(x)x^{n/2} + B_0(x)$$

$$A(x) * B(x) = A_1(x) * B_1(x)x^n + (A_0(x) * B_1(x) + A_1(x) * B_0(x))x^{n/2} + A_0(x) * B_0(x)$$

lo cual a primera vista requeriría 4 multiplicaciones recursivas, pero de manera ingeniosa se lograba reducir este número a sólo 3, obteniéndose un algoritmo de tiempo  $\Theta(n^{\log_2 3}) = \Theta(n^{1.59})$ . Si esto mismo se hace dividiendo cada polinomio en *tercios* en lugar de mitades, de manera trivial se requerirían 9 multiplicaciones recursivas, pero este número también se puede reducir usando técnicas similares. ¿A cuánto debería reducirse este número de multiplicaciones recursivas para que el algoritmo resultante sea más eficiente que el visto en clases? Justifique su respuesta usando el Teorema Maestro.

- b) Resuelva la ecuación de recurrencia:

$$a_n = a_{n-1} + 6a_{n-2}$$

con  $a_0 = 0, a_1 = 5$ .

2. Escriba un método que reciba como parámetro un String que contiene el texto de otro programa en Java, y retorne como resultado otro String que contenga el mismo programa, al cual se le han eliminado todos sus comentarios. Considere sólo comentarios que comienzan con “//” y se extienden hasta el final de la línea. Suponga que cada fin de línea se representa por un carácter *newline* (`'\n'`). Tenga cuidado que un // que aparezca dentro de una constante de tipo String no debe ser confundido con el comienzo de un comentario. Para simplificar, suponga que dentro de las comillas no hay *newlines* ni otras comillas. Diseñe primero un diagrama de estados y a partir de él escriba su programa.

3.

- a) Escriba un método booleano llamado `EsJeraquico`, que reciba como parámetro un puntero a la raíz de un árbol binario, y retorne verdadero si se cumple la siguiente condición: para todos los nodos, el valor en su campo de información debe ser mayor o igual que los valores almacenados en sus hijos. Si cualquier nodo tiene un valor mayor que el de su padre, debe retornarse falso. Utilice la siguiente definición:

```
class NodoArbol
{
    int info;
    NodoArbol izq;
    NodoArbol der;
}
```

- b) Suponga que se tiene una lista enlazada con nodos definidos por

```
class NodoLista
{
    int info;
    NodoLista sgte;
}
```

donde la variable `lista` apunta al comienzo de la lista, la cual se inicia con un nodo cabecera. Usted debe escribir un método llamado `BuscarYMover`, que reciba un parámetro de tipo `int` llamado `x`. Este método debe buscar en la lista un nodo que contenga el valor `x`. Si lo encuentra, debe adelantarlo un lugar en la lista, excepto si ya estaba en primer lugar, en cuyo caso debe dejarlo donde está.

4. Suponga que se dispone de un arreglo  $a[0], \dots, a[n-1]$ , que contiene números enteros desordenados, que pueden ser positivos o negativos. Escriba un trozo de programa que reordene estos datos dejando alternados valores positivos y negativos, comenzando con uno positivo. Si hay más de un tipo que del otro, los que sobran deben quedar al extremo derecho del arreglo. El proceso debe demorar a lo más tiempo proporcional al tamaño del arreglo ( $O(n)$ ) y debe efectuarse usando sólo el arreglo  $a$  proporcionado, sin usar otro arreglo auxiliar. Encuentre un invariante apropiado, descríballo claramente (use un dibujo para ello) y explique su programa en términos de dicho invariante.

*Ejemplo:*

ANTES: 

-15	17	42	60	-32	21	13	-44	-72	24	-25	37
-----	----	----	----	-----	----	----	-----	-----	----	-----	----

DESPUÉS: 

17	-15	42	-32	60	-44	13	-72	21	-25	24	37
----	-----	----	-----	----	-----	----	-----	----	-----	----	----

Tiempo: 2:00 horas

Entregar en hojas separadas

Con apuntes de clases