

# CC30A Algoritmos y Estructuras de Datos, Secciones 1 y 2

## Control 1

Prof. Patricio Poblete, Benjamin Bustos

Fecha: 3 de mayo de 2002.

Tiempo: 2 horas.

### Pregunta 1

- i. Analice el número de comparaciones que realiza el siguiente algoritmo en el peor caso. El programa se invoca inicialmente como `funcion(A, x, 0, n-1)` para buscar el número  $x$  en el arreglo  $A[0..n-1]$ , el cual se encuentra ordenado de menor a mayor:

```
int funcion(int[] a, int x, int i, int j) {
    if (i>j)
        return -1;
    if (i==j) {
        if (a[i]==x)
            return i;
        else
            return -1;
    }
    else {
        int n1=(2i+j)/3;
        int n2=(i+2j)/3;
        if (x<=a[n1])
            j=n1;
        else if (x>=a[n2])
            i=n2;
        else {
            i=n1+1;
            j=n2-1;
        }
        return funcion(a, x, i, j);
    }
}
```

Primero escriba la ecuación de recurrencia respectiva y luego resuélvala. Suponga en ambos casos que  $n$  es de la forma  $n = 3^k, k \in \mathbb{N}$ .

ii. Resuelva la siguiente ecuación de recurrencia:

$$a_{n+2} - 5a_{n+1} + 6a_n = 0, a(0) = 0, a(1) = 1$$

## Pregunta 2

Implemente en Java el método `void cambiarPrioridad(int i, int nuevaPrioridad)` perteneciente a la clase `HeapBinario`, que modifica la prioridad el elemento  $i$ -ésimo del heap a *nuevaPrioridad*. La nueva prioridad del elemento  $i$ -ésimo puede ser tanto mayor o menor que la prioridad anterior. Note que al cambiar la prioridad del elemento se puede perder la propiedad de orden de prioridad en el heap, en cuyo caso su implementación debe recuperarla.

## Pregunta 3

Implemente en Java el método `int alturaArbol(Nodo raíz)`, que recibe por parámetro la raíz de un árbol binario y retorna la altura de éste. La clase `Nodo` está definida como:

```
class Nodo {
    Object info;
    Nodo izq;
    Nodo der;
}
```

## Pregunta 4

Modifique el algoritmo de Dijkstra, el cual dado un grafo  $G = (V, E)$  y un nodo de partida  $s \in V$  encuentra los caminos más cortos entre  $s$  y todo otro nodo  $v \in V$ , para que adicionalmente cuente el número de caminos de costo mínimo entre el nodo inicial  $s$  y los restantes, si es que existe más de un camino con costo mínimo. Explique cómo debe modificarse el algoritmo y luego impleméntelo en pseudocódigo.