



IIC1253 - Sección 1 - Segundo Semestre 2014

Profesor: Marcelo Arenas

Ayudantes: Matías San Martín (*masanmartin@uc.cl*) - Martín Muñoz (*mmunos@uc.cl*)

## Ayudantía 15.

*Análisis de Algoritmos.*

**Problema 1.** Sea  $p : \mathbb{N} \rightarrow \mathbb{R}_0^+$  un polinomio de grado  $k \in \mathbb{N}$  con coeficientes reales, dado por

$$p(n) = a_0 + a_1n + a_2n^2 + \cdots + a_kn^k, \quad \text{para todo } n \in \mathbb{N} \text{ y } a_k \neq 0.$$

Determine el único valor de  $i \in \mathbb{N}$  tal que  $p \in \Theta(n^i)$ . Demuestre que efectivamente lo es.

**Problema 2.** Considere el siguiente algoritmo, definido para todo  $n \in \mathbb{N} \setminus \{0\}$ , dado por:

```

1: function ALGORITMO(int n)
2:   s ← 0
3:   for i = 1 .. n do
4:     for j = 1 .. i do
5:       s ← s + j · (i - j + 1)
6:     end for
7:   end for
8:   return s
9: end function

```

Sea  $f : \mathbb{N} \rightarrow \mathbb{N}$  la función que entrega la cantidad de sumas, restas y multiplicaciones realizadas al ejecutar ALGORITMO( $n$ ), con  $n \in \mathbb{N} \setminus \{0\}$ . Encuentre una función  $g$ , tal que  $f \in \Theta(g)$ .

**Problema 3.** Sea  $F : \mathbb{N} \rightarrow \mathbb{N}$  definida como la sucesión de Fibonacci, es decir, que

$$F(n) = \begin{cases} n & \text{si } n = 0 \text{ o } n = 1 \\ F(n-1) + F(n-2) & \text{si } n \geq 2 \end{cases} \quad \text{para todo } n \in \mathbb{N}.$$

Encuentre una constante  $c \in \mathbb{R}^+$ , con  $c > 1$ , tal que  $F \in \Theta(c^n)$ . Demuestre.

**Problema 4.** Considere los siguientes algoritmos recursivos que calculan el valor de  $x^n$ , para  $x, n \in \mathbb{N}$ :

---

```

1: function ELEVAR1(int  $x$ , int  $n$ )
2:   if  $n = 0$  then
3:     return 1
4:   else
5:     return  $x \cdot \text{ELEVAR1}(x, n - 1)$ 
6:   end if
7: end function

```

---

```

1: function ELEVAR2(int  $x$ , int  $n$ )
2:   if  $n = 0$  then
3:     return 1
4:   else if  $n = 1$  then
5:     return  $x$ 
6:   else if  $n \equiv 0 \pmod{2}$  then
7:     return  $\text{ELEVAR2}(x, n/2) \cdot \text{ELEVAR2}(x, n/2)$ 
8:   else
9:     return  $x \cdot \text{ELEVAR2}(x, (n - 1)/2) \cdot \text{ELEVAR2}(x, (n - 1)/2)$ 
10:  end if
11: end function

```

---

Para cada uno de estos algoritmos, realice lo siguiente considerando un  $x \in \mathbb{N}$  fijo:

- Establezca una recurrencia para  $T_x(n)$ , donde  $T_x(n)$  corresponde a la cantidad de comparaciones realizadas por el programa con input  $x, n \in \mathbb{N}$  en el peor caso.
- Resuelva la recurrencia encontrada en la parte anterior, encontrando una expresión cerrada para  $T_x(n)$  definida para todo  $n \in \mathbb{N}$  y determine una función  $f$  tal que  $T_x \in \Theta(f)$ .

**Problema 5.** Sea  $T : \mathbb{N} \setminus \{0\} \rightarrow \mathbb{R}_0^+$  dada por la siguiente recurrencia para  $n \in \mathbb{N} \setminus \{0\}$ , con  $a, b, c \in \mathbb{R}_0^+$  y  $b \geq 2$ .

$$T(n) = \begin{cases} aT(\lceil n/b \rceil) + n^c & \text{si } n > 1 \\ 1 & \text{si } n = 1 \end{cases} \quad \text{para todo } n \in \mathbb{N} \setminus \{0\}.$$

Resuelva la recurrencia para  $T$  y muestre que si  $\log_b(a) < c$ , entonces  $T \in \Theta(n^c)$ .