Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ciencias Exactas y Naturales
Departamento de Computación
Estructura de Datos

Práctica 1
Notación Asintótica, Sumatorias y Relaciones de Recurrencia

1. Demostrar el siguiente teorema,

Teorema 1

$f \in \Theta(g)$ si y solo si existen constantes c_1, $c_2 \in \Re^+$, $n_0 \in$ N tales que

$$\forall n \geq n_0 \cdot 0 \leq c_1 \mathrm{g(n)} \leq \mathrm{f(n)} \leq c_2 \mathrm{g(n)}$$

2. Dadas f, g : N $\to$ R, demostrar las siguientes propiedades de las notaciones asintóticas:

1. $\mathrm{f(n)} = \mathrm{O(g(n))} \wedge \mathrm{g(n)} = \mathrm{O(h(n))} \Rightarrow \mathrm{f(n)} = \mathrm{O(h(n))}$
2. $\mathrm{f(n)} = \Omega\mathrm{(g(n))} \wedge \mathrm{g(n)} = \Omega\mathrm{(h(n))} \Rightarrow \mathrm{f(n)} = \Omega\mathrm{(h(n))}$
3. f asintóticamente no negativa $\Rightarrow \mathrm{f(n)} = \Theta\mathrm{(f(n))}$
4. $\mathrm{f(n)} = \Theta\mathrm{(g(n))} \Leftrightarrow \mathrm{g(n)} = \Theta\mathrm{(f(n))}$
5. $\mathrm{f(n)} = \mathrm{O(g(n))} \Leftrightarrow \mathrm{g(n)} = \Omega\mathrm{(f(n))}$
6. $\mathrm{f(n)} = \mathrm{O(g(n))} \Rightarrow \forall k \in \Re^+ \cdot \mathrm{k\ f(n)} = \mathrm{O(g(n))}$
7. $\mathrm{f(n)} = \Omega\mathrm{(g(n))} \Rightarrow \forall k \in \Re^+ \cdot \mathrm{k\ f(n)} = \Omega\mathrm{(g(n))}$

3. Sean f, g : N $\to$ R asintóticamente no negativas y h(n) = max(f(n), g(n)), demostrar que

$$\mathrm{h(n)} = \Theta\mathrm{(f(n) + g(n))}$$

4. Sean a, b $\in \Re$ constantes, b positivo, probar que

a. $(n+a)^b = \Theta(n^b)$

b. $b^n = \Theta(b^{(}n+a))$

5. Demostrar el siguiente teorema,

Teorema 2

Sean f, g : N $\to \Re$ asintóticamente no negativas y $k \in \Re^+$, si

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = k$$

,entonces f(n) = Θ(g(n)).

6. Sea

$$\sum_{i=0}^{d} a_i n^i$$

un polinomio de grado d asintóticamente positivo (a_d positivo). Demostrar que p(n) = $\Theta(n^d)$.

7. Probar que

- lg(n!) = $\Theta(n \lg n)$
- $\sum_{k=1}^{n} k^2 = \Theta(n^3)$
- $\sum_{k=1}^{n} \frac{1}{k^2} = \mathrm{O}(1)$

8. Las Torres de Hanoi

El juego de las Torres de Hanoi consiste en n discos de diferentes diámetros que pueden colocarse en tres varillas verticales. Inicialmente, los discos se encuentran colocados en orden creciente de diámetro en la primer varilla, como muestra la Fig. 1. El objetivo es obtener la configuración final a la derecha de la figura, moviendo un disco a la vez de encima de una varilla a otra y nunca colocando un disco de un determinado diámetro sobre otro de un diámetro menor.

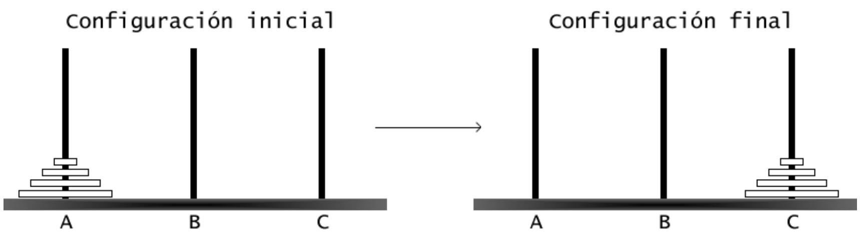

Figura 1: Configuraciones inicial y final del problema de las Torres de Hanoi para $n = 4$.

¿Si tuviéramos un algoritmo que resuelva el juego de las Torres de Hanoi, cual sería su complejidad temporal? Plantear la recurrencia y cerrarla.

Ayuda: Tener en cuenta la forma en que se mueven los discos.

9. Suponga que para definir la multiplicación de dos números naturales solo dispone del operador de adición, de un operador doble para duplicar un número y otro operador mitad para calcular la división entera de un número por 2. Se pueden multiplicar dos números en forma eficiente con estos operadores usando la siguiente técnica:

- Escribir los números en dos columnas.
- Sucesivamente dividir por dos el número en la primer columna y duplicar el número en la segunda columna hasta que el número en la primer columna sea 1.
- Sumar los números en la segunda columna de todas las filas donde el número en la primer columna sea impar. El resultado es el producto de los dos números originales.

Ejemplo: 22×43

22	43
11	**86**
5	**172**
2	344
1	**688**
	946

1. En base a la especificación del ejercicio anterior escribir una función recursiva.
2. Calcular su complejidad temporal.