Fundamentos de Programación (FdP)
Taller No. 00 (FADA-Formulas)
Entrega: Solamente a través del Campo Virtual.

Antonio J. Vélez Q.
antonio.velez@correounivalle.edu.co
Universidad del Valle – Sede Palmira

Notación Asintótica - Definición

Notación Θ

Se dice que f(n) es Θ (g(n)) sii existe c_1, c_2, $n_0 > 0$ tal que $c_1 g(n) <= f(n) <= c_2 g(n)$ para todo $n > n_0$

Notación O

Se dice que f(n) es O(g(n)) sii existe c_1, $n_0 > 0$ tal que $0 <= f(n) <= c_1 g(n)$ para todo $n > n_0$

Notación Ω

Se dice que f(n) es Ω (g(n)) sii existe c_1, $n_0 > 0$ tal que $0 <= c_1 g(n) <= f(n)$ para todo $n > n_0$

Notación Asintótica - Propiedades

Transitividad

$$f(n) \in \Theta(g(n)) \quad \wedge \quad g(n) \in \Theta(h(n)) \quad \textit{imply } f(n) \in \Theta(h(n))$$

$$f(n) \in O(g(n)) \quad \wedge \quad g(n) \in O(h(n)) \quad \textit{imply } f(n) \in O(h(n))$$

$$f(n) \in \Omega(g(n)) \quad \wedge \quad g(n) \in \Omega(h(n)) \quad \textit{imply } f(n) \in \Omega(h(n))$$

Reflexividad

$$f(n) \in \Theta(f(n))$$
$$f(n) \in O(f(n))$$
$$f(n) \in \Omega(f(n))$$

Simetría

$$f(n) \in \Theta(g(n)) \quad \text{if and only if} \quad g(n) \in \Theta(f(n))$$

Simetría Transpuesta

$$f(n) \in O(g(n)) \quad \text{if and only if} \quad g(n) \in \Omega(f(n))$$

$$f(n) \in o(g(n)) \quad \text{if and only if} \quad g(n) \in \omega(f(n))$$

Similitud (con valores numéricos)

$$f(n) \in O(g(n)) \quad \textit{similar a} \quad a \le b$$
$$f(n) \in \Omega(g(n)) \quad \textit{similar a} \quad a \ge b$$
$$f(n) \in \Theta(g(n)) \quad \textit{similar a} \quad a = b$$
$$f(n) \in o(g(n)) \quad \textit{similar a} \quad a < b$$
$$f(n) \in \omega(g(n)) \quad \textit{similar a} \quad a > b$$

Sumatorias - Propiedades

$$\sum_{k=1}^{n} (c a_k) = c \sum_{k=1}^{n} (a_k)$$

$$\sum_{k=0}^{n} \frac{a^k}{b^k} = \sum_{k=0}^{n} \left(\frac{a}{b}\right)^k$$

$$\sum_{k=1}^{n} (a_k + b_k) = \sum_{k=1}^{n} (a_k) + \sum_{k=1}^{n} (b_k)$$

$$\sum_{k=1}^{n} \Theta(F(k)) = \Theta\left(\sum_{k=1}^{n} F(k)\right)$$

$$\sum_{k=a}^{b} a_k = \sum_{k=1}^{b} a_k - \sum_{k=1}^{a-1} a_k$$

Sumatorias – Soluciones

$$\sum_{k=0}^{n} a r^k = \frac{a r^{n+1} - a}{r - 1}$$
$$r \neq (0, 1) \ ; \ a \, constante$$

$$\sum_{k=0}^{n} x^k = \frac{x^{n+1} - 1}{x - 1} \qquad \sum_{k=0}^{\infty} x^k = \frac{1}{1 - x} \quad para |x| < 1$$

$$\sum_{k=1}^{\infty} k\, x^{k-1} = \frac{1}{(1 - x)^2} \quad para |x| < 1$$

$$\sum_{k=0}^{n} 2^k = 2^{n+1} - 1$$

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2} \qquad \sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4} \qquad \sum_{k=1}^{n} \frac{1}{2^k} = 1 - \frac{1}{2^n}$$

Exponenciales

$$a^0 = 1 \qquad a^1 = a \qquad a^m * a^n = a^{m+n}$$

$$(a^m)^n = a^{m.n} = a^{n.m} = (a^n)^m$$

$$a^{-1} = \frac{1}{a} \qquad \sqrt{a} = a^{1/2}$$

Logaritmos

$$\log_b^k a = (\log_b a)^k \qquad \log\ \log n = \log(\log n)$$

$$a = b^{\log_b a} \qquad \log_b b = 1$$

$$\log_c(a\,b) = \log_c a + \log_c b$$

$$\log_b a^n = n \log_b a \qquad a^{\log_b n} = n^{\log_b a}$$

$$\log_b a = \frac{\log_c a}{\log_c b} \qquad \log_b \frac{1}{a} = -\log_b a$$

Árbol de Recurrencia

Para una recursión de la forma

T(n) = aT(n/b) + f(n)

El valor de los nodos internedios es f(n/b^i), del cual se desprenden '**a**' subploblemas de tamaño '**n/b**' (T(n/b))

La altura del árbol es log $_b$ n (recuerde que n/b es el tamaño de los subproblemas)

El valor de las hojas del árbol es T(1); habiendo un total de $a^{\log_b n}$ hojas ($n^{\log_b a}$ aplicando las propiedades de los logaritmos)

Método Maestro

Sea a>=1, b>1 valores constantes, f(n) una función y T(n) definida en los enteros no negativos por la recurrencia

T(n) = aT(n/b) + f(n)

1. Si f(n) = $O(n^{\log_b a - \epsilon})$ para una constante $\epsilon > 0$, entonces

   T(n) = $\Theta(n^{\log_b a})$

2. Si f(n) = $\Theta(n^{\log_b a})$ entonces

   T(n) = $\Theta(n^{\log_b a} . \log_b n)$

3. Si f(n) = $\Omega(n^{\log_b a + \epsilon})$ para una constante $\epsilon > 0$ y si $a\, f(n/b) \le c\, f(n)$, para (c < 1) y para un n suficientemente grande, entonces

   T(n) = $\Theta(f(n))$