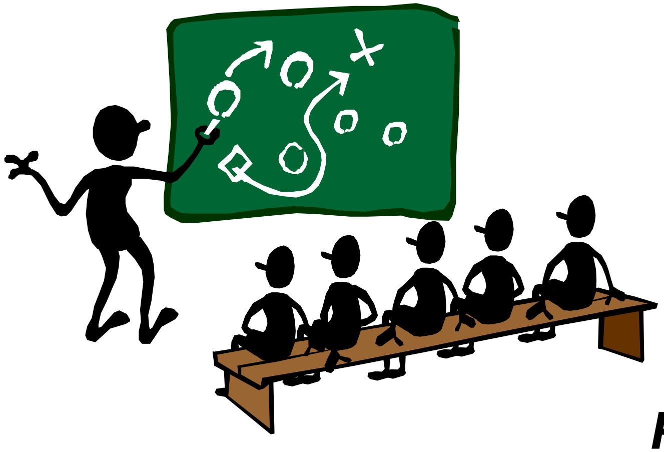
Algorithms – Chapter 3
Growth of Functions



Juinn-Dar Huang

Professor

jdhuang@mail.nctu.edu.tw

July 2007

Rev. ’08, ’11, ’12, ’15, ’16, ’18, ’19, ’20, ’21

Asymptotic Notations

- Though the exact running time can sometimes be calculated…
  - as we did for insertion sort
- The extra precision is not usually worth the effort
- Alternatively, study the asymptotic efficiency of algorithms
  - for a large input size
- Usually, an algorithm that is asymptotically more efficient will be the best choice for all but very small inputs

Juinn-Dar Huang
jdhuang@mail.nctu.edu.tw

Juinn-Dar Huang
jdhuang@mail.nctu.edu.tw

Θ-Notation

- For a given function g(n), Θ(g(n)) denotes the **set** of functions:

$$\Theta(g(n)) = \{ f(n) \mid \exists c_1, c_2, n_0 \text{ s.t. } 0 \le c_1 g(n) \le f(n) \le c_2 g(n) \text{ for all } n \ge n_0 \}$$

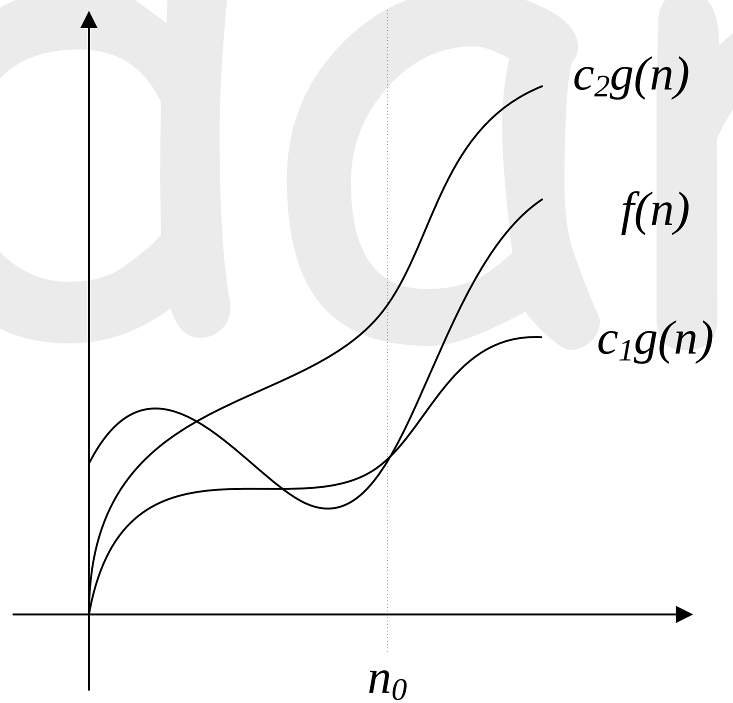
$f(n) \in \Theta(g(n))$ → **Standard set notation**

$f(n) = \Theta(g(n))$ → **Abuse; but commonly used**

- g(n) is an **asymptotically tight bound** for f(n)

Illustration

- The definition of $\Theta(g(n))$ requires that
  - f(n) must be **asymptotically nonnegative**
  - consequently, g(n) must be **asymptotically nonnegative**, too



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

Examples

- Show that $\frac{1}{2}n^2 - 3n = \Theta(n^2)$

$$\Rightarrow \frac{n^2}{14} \le \frac{n^2}{2} - 3n \le \frac{n^2}{2} \; if \;\; n > 7.$$

- $6n^3 \neq \Theta(n^2)$

- Polynomials

$$p(n) = \sum_{i=0}^{d} a_i n^i \text{ where } a_i \text{ are constant with } a_d > 0.$$

$$\text{Then } P(n) = \Theta(n^d).$$

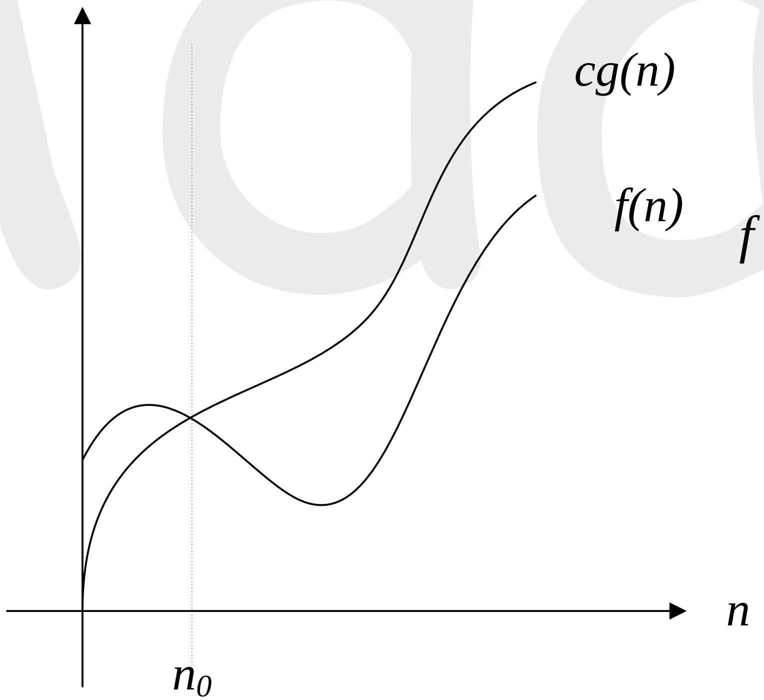
Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

O-Notation (1/2)

- For a given function g(n), O(g(n)) denotes the set of functions:

  $$O(g(n)) = \{ f(n) \mid \exists c, n_0 \text{ s.t. } 0 \le f(n) \le cg(n)\ \forall n \ge n_0 \}$$

- g(n) is an **asymptotically upper bound** for f(n)



$f(n) = O(g(n))$

Juinn-Dar Huang
jdhuang@mail.nctu.edu.tw

O-Notation (2/2)

- If f(n) = Θ(g(n)), then f(n) = O(g(n))
- That is, Θ(g(n)) ⊆ O(g(n))

- As well

$$p(n) = \sum_{i=0}^{d} a_i n^i \text{ where } a_i \text{ are constant with } a_d > 0.$$

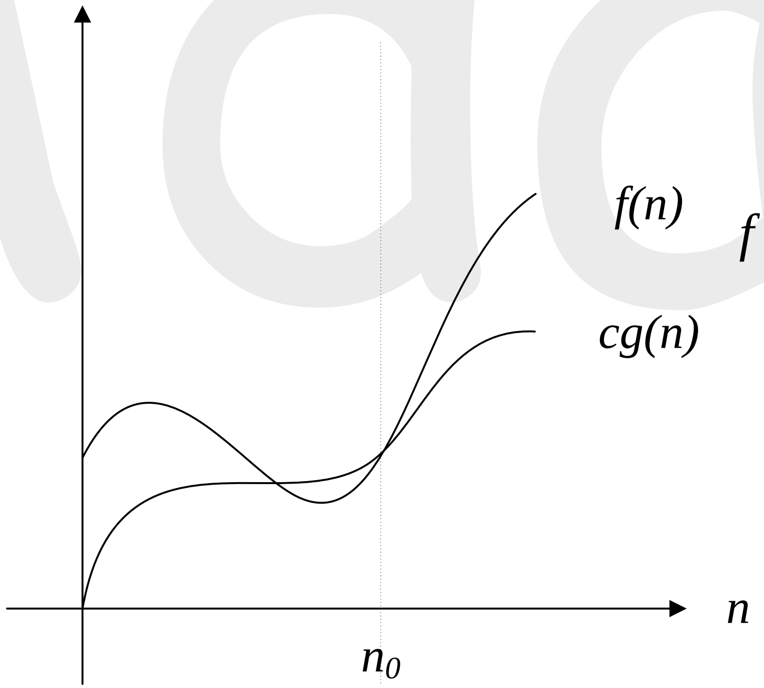
Then $P(n) = O(n^d)$.

Ω-Notation

- For a given function g(n), Ω(g(n)) denotes the set of functions:

$$\Omega(g(n)) = \{ f(n) \mid \exists c, n_0 \text{ s.t. } 0 \le cg(n) \le f(n) \ \forall n \ge n_0 \}$$

- g(n) is an **asymptotically lower bound** for f(n)



$$f(n) = \Omega(g(n))$$

Theorem

- For any two functions *f*(*n*) and *g*(*n*), $f(n)=\Theta(g(n))$ if and only if $f(n)=O(g(n))$ and $f(n)=\Omega(g(n))$

Juinn-Dar Huang
jdhuang@mail.nctu.edu.tw

Asymptotic Notations in Equations

- $n = O(n^2)$ ⇒ actually means $n \in O(n^2)$

- $2n^2+3n+1 = 2n^2 + \Theta(n)$
  - i.e., $2n^2+3n+1 = 2n^2 + f(n)$ where $f(n) \in \Theta(n)$

- $2n^2 + \Theta(n) = \Theta(n^2)$
  - $\forall\ f(n) \in \Theta(n)$,
  - $\exists\ g(n) \in \Theta(n^2)$ such that
  - $2n^2 + f(n) = g(n)$ for all n

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

o-Notation

- O-notation may or may not be asymptotically tight
  - $2n^2 = O(n^2)$ ➔ asymptotically tight
  - $2n = O(n^2)$ ➔ not asymptotically tight
  - o-notation is used to denote an upper bound that is not asymptotically tight

$$o(g(n)) = \{ f(n) \mid \underline{\forall c > 0}, \exists n_0 > 0 \text{ s.t. } \forall n \geq n_0, 0 \leq f(n) < cg(n) \}$$

$$f(n) = o(g(n)) \Leftrightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

- Examples
  - $2n = o(n^2)$
  - $2n^2 \neq o(n^2)$

ω-Notation

- By analogy
  - O-notation vs. o-notation ⇔ Ω-notation vs. ω-notation

$$\omega(g(n)) = \{ f(n) \mid \underline{\forall c > 0}, \exists n_0 > 0 \text{ s.t. } \forall n \geq n_0, 0 \leq cg(n) < f(n) \}$$

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

- Examples
  - $n^2/2 = \omega(n)$
  - $n^2/2 \neq \omega(n^2)$

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Growth of Functions

Properties (1/2)

- Transitivity

$$f(n) = \Theta(g(n)) \wedge g(n) = \Theta(h(n)) \Rightarrow f(n) = \Theta(h(n))$$
$$f(n) = O(g(n)) \wedge g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$$
$$f(n) = \Omega(g(n)) \wedge g(n) = \Omega(h(n)) \Rightarrow f(n) = \Omega(h(n))$$
$$f(n) = o(g(n)) \wedge g(n) = o(h(n)) \Rightarrow f(n) = o(h(n))$$
$$f(n) = \omega(g(n)) \wedge g(n) = \omega(h(n)) \Rightarrow f(n) = \omega(h(n))$$

- Reflexivity

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

Properties (2/2)

- Symmetry

$$f(n) = \Theta(g(n)) \Leftrightarrow g(n) = \Theta(f(n))$$

- Transpose symmetry

$$f(n) = O(g(n)) \Leftrightarrow g(n) = \Omega(f(n))$$

$$f(n) = o(g(n)) \Leftrightarrow g(n) = \omega(f(n))$$

Monotonicity

- Monotonicity
  - A function f is *monotonically increasing (non-decreasing)* if $m < n$ implies $f(m) \leq f(n)$
  - A function f is *monotonically decreasing (non-increasing)* if $m < n$ implies $f(m) \geq f(n)$
  - A function f is *strictly increasing* if $m < n$ implies $f(m) < f(n)$
  - A function f is *strictly decreasing* if $m < n$ implies $f(m) > f(n)$

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Floors and Ceilings

$$x - 1 < \lfloor x \rfloor \le x \le \lceil x \rceil < x + 1$$

$$\lceil n/2 \rceil + \lfloor n/2 \rfloor = n$$

$$\lceil \lceil x/a \rceil / b \rceil = \lceil x/ab \rceil$$

$$\lfloor \lfloor x/a \rfloor / b \rfloor = \lfloor x/ab \rfloor$$

$$\lceil a/b \rceil \le (a + (b-1))/b$$

$$\lfloor a/b \rfloor \ge (a - (b-1))/b$$

x: real number
n: integer
a, b: integral constants

Juinn-Dar Huang
jdhuang@mail.nctu.edu.tw

Growth of Functions

Modular Arithmetic

- For any integer a and any positive integer n, the value a mod n is the **remainder** (or **residue**) of the quotient a/n :
  - $a \bmod n = a - \lfloor a/n \rfloor * n$
  - If(a mod n) = (b mod n). We write $a \equiv b \pmod{n}$ and say that a is **equivalent** to b, modulo n
  - We write $a \not\equiv b \pmod{n}$ if a is not equivalent to b, modulo n

Juinn-Dar Huang
jdhuang@mail.nctu.edu.tw

Polynomials

$$p(n) = \sum_{i=0}^{d} a_i n^i \text{ where } a_i \text{ are constant with } a_d > 0.$$

Then $P(n) = \Theta(n^d)$.

- A function f(n) is polynomially bounded if f(n) = $O(n^k)$ for some constant k

Exponentials

- For all real constants a > 1 and b
  - $\lim_{n\to\infty} \frac{n^b}{a^n} = 0$
  - ➔ $n^b = o(a^n)$

Logarithms

- Notations
  - $\lg n = \log_2 n$
  - $\ln n = \log_e n$
  - $\lg^k n = (\lg n)^k$
  - $\lg \lg n = \lg(\lg n)$
- A function f(n) is polylogarithmically bounded if $f(n) = O(\lg^k n)$ for some constant k
- **$\lg^b n = o(n^a)$**

$$\lim_{n\to\infty} \frac{n^b}{a^n} = 0$$

for all real constant a > 1 and b

Substitute lg*n* for n and 2^a for a

$$\lim_{n\to\infty} \frac{\lg^b n}{(2^a)^{\lg n}} = \lim_{n\to\infty} \frac{\lg^b n}{n^a} = 0$$

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Factorials

- Weak upper bound: $n! \le n^n$

- Properties
  - $n! = o(n^n)$

    $n! = \omega(2^n)$

    $\lg(n!) = \Theta(n \lg n)$

Functional Iterations

$$f^{(i)}(n) = \begin{cases} n & \text{if } i = 0, \\ f(f^{(i-1)}(n)) & \text{if } i > 0. \end{cases}$$

For example, if $f(n) = 2n$, then $f^{(i)}(n) = 2^i n$

Iterated Logarithm Functions

$$\lg^{(i)}(n) = \begin{cases} n & \text{if } \ i = 0 \\ \lg(\lg^{(i-1)} n) & \text{if } \ i > 0 \ \text{ and } \ \lg^{(i-1)} n > 0 \\ \text{undefined} & \text{if } \ i > 0 \ \text{ and } \ \lg^{(i-1)} n \le 0 \\ & \quad \text{or } \lg^{(i-1)} n \ \text{ is undefined.} \end{cases}$$

$$\lg^{*}(n) = \min\{\, i \ge 0 \;:\; \lg^{(i)} n \le 1 \,\}$$

$$lg^{*}\ 2 = 1$$

$$lg^{*}\ 4 = 2$$

$$lg^{*}\ 16 = 3$$

$$lg^{*}\ 65536 = 4$$

$$lg^{*}\ 2^{65536} = 5$$

Fibonacci Numbers (1/2)

- Fibonacci numbers
  - $F_0 = 0, F_1 = 1, F_i = F_{i-1} + F_{i-2}$ for $i \geq 2$
  - 0, 1,1, 2, 3, 5, 8, 13, …
- Golden ratio
  - $\phi = \dfrac{1+\sqrt{5}}{2} = 1.61803\ldots$ **one of the roots of $x^2 - x - 1 = 0$**
  - $\hat{\phi} = \dfrac{1-\sqrt{5}}{2} = -0.61803\ldots$ **another root of $x^2 - x - 1 = 0$**

Fibonacci Numbers (2/2)

$$F_i = \frac{\phi^i - \widehat{\phi}^i}{\sqrt{5}}$$

Try to prove it by math induction

$$\because \quad \left|\hat{\phi}\right| < 1 \Rightarrow \frac{\left|\hat{\phi}^i\right|}{\sqrt{5}} < \frac{1}{\sqrt{5}} < \frac{1}{2}$$

F_i = $\frac{\phi^i}{\sqrt{5}}$ rounded to the nearest integer

The Fibonacci sequence grows exponentially !

Growth of Functions

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Function Names

1	constant
$\lg^* n$	iterated logarithm
$\lg^{(O(1))} n = \underbrace{\lg \lg \ldots \lg}_{O(1)} n$	—
$\lg n$	logarithmic
$\lg^{O(1)} n = (\lg n)^{O(1)}$	polylogarithmic
$\sqrt{n}$	sublinear
n	linear
$n \lg n$	loglinear
n^2	quadratic
n^3	cubic
n^4	quartic
$2^n, 3^n, \ldots$	exponential
$n!$	factorial
n^n	-

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Growth of Functions

Growth of Functions

Assume 1 BIPS, 1 instruction/operation

Order	Θ	$n=10$	$n=100$	$n=10^3$	$n=10^6$
1	$\Theta(1)$	1×10^{-9} sec	1×10^{-9} sec	1×10^{-9} sec	1×10^{-9} sec
$\lg^* n$	$\Theta(\lg^* n)$	3×10^{-9} sec	3×10^{-9} sec	3×10^{-9} sec	4×10^{-9} sec
$\lg \lg n$	$\Theta(\lg \lg n)$	2×10^{-9} sec	3×10^{-9} sec	3×10^{-9} sec	4×10^{-9} sec
$\lg n$	$\Theta(\lg n)$	3×10^{-9} sec	7×10^{-9} sec	1×10^{-8} sec	2×10^{-8} sec
$\sqrt{n}$	$\Theta(\sqrt{n})$	3×10^{-9} sec	1×10^{-8} sec	3×10^{-8} sec	1×10^{-6} sec
n	$\Theta(n)$	1×10^{-8} sec	1×10^{-7} sec	1×10^{-6} sec	0.001 sec
$n \lg n$	$\Theta(n \lg n)$	3×10^{-8} sec	2×10^{-7} sec	3×10^{-6} sec	0.006 sec
n^2	$\Theta(n^2)$	1×10^{-7} sec	1×10^{-5} sec	0.001 sec	16.7 min
n^3	$\Theta(n^3)$	1×10^{-6} sec	0.001 sec	1 sec	3×10^{5} cent.
2^n	$\Theta(2^n)$	1×10^{-6} sec	3×10^{17} cent.	∞	∞
$n!$	$\Theta(n!)$	0.003 sec	∞	∞	∞

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw