Growth of Functions

Asymptotic Notation

- 충분히 큰 n 에 대하여, 함수 f(n) 을 근사적으로 표현하는 방법
  - 알고리즘의 running time 을 표현하기 위하여 사용

- 종류
  - Θ : asymptotic tight bound
  - O : asymptotic lower bound
  - Ω : asymptotic upper bound
  - o : asymptotic strict lower bound
  - ω : asymptotic strict upper bound

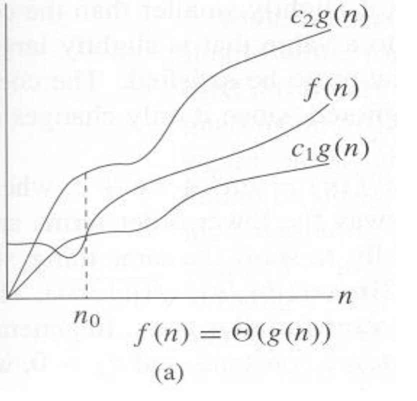
Asymptotic Notation

- **Θ** notation
  - Definition

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

$\leftrightarrow$ there exist positive constants c_1, c_2, n_o such that

$$0 \le c_1 g(n) \le f(n) \le c_2 g(n) \qquad \forall n \ge n_o$$



- Asymptotically tight bound

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

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

$2n \neq \Theta(n^2)$

Asymptotic Notation

- ***O*** notation
  - Definition

$f(n) = O(g(n))$
$\leftrightarrow$ there exist positive constants c, n_o such that

$$0 \le f(n) \le cg(n) \qquad \forall n \ge n_o$$

- Asymptotic upper bound
  Worst-case 의 running time 표현에 적합

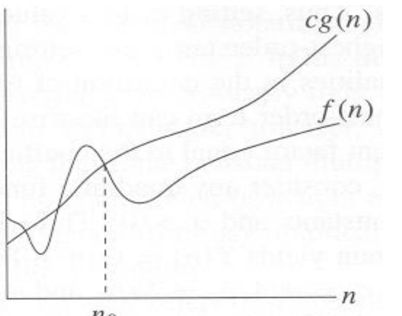
-

$$\Theta(g(n)) \subseteq O(g(n))$$

(Ex)) $an^2 + bn + c = O(n^2)$

$an^3 \neq O(n^2)$

$an + b = O(n^2)$



f(n) = O(g(n))
(b)

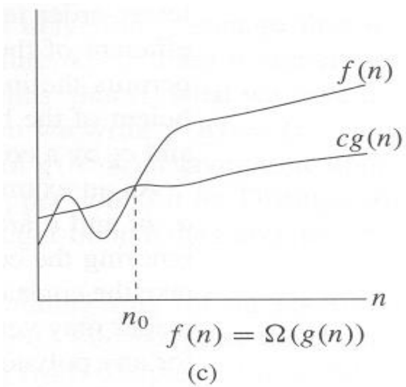
Asymptotic Notation

- ***Ω*** notation
  - Definition

$f(n) = \Omega(g(n))$
$\leftrightarrow$ there exist positive constants c, n_o such that

$$0 \le cg(n) \le f(n) \qquad \forall n \ge n_o$$

  - Asymptotical lower bound
    Best-case 의 running time 표현에 적합



(Ex) $an^2 + bn + c = \Omega(n^2)$

$an^2 + bn + c = \Omega(n)$

$an + b \neq \Omega(n^2)$

(Ex) Insertion Sort 알고리즘의 running time: $\Omega(n) \sim O(n^2)$

Asymptotic Notation

- ***o*** notation
  - Definition

$f(n) = o(g(n))$

$\leftrightarrow$ there exist positive constants c, n_o such that

$$0 \le f(n) < cg(n)$$

- Asymptotic strict upper bound

  Upper bound that is not asymptotically tight

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

(Ex) $an^2 + bn + c \neq o(n^2)$

$an + b = o(n^2)$

Asymptotic Notation

- $\boldsymbol{\omega}$ notation
  - Definition

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

$\leftrightarrow$ there exist positive constants $\boldsymbol{c}, \boldsymbol{n_o}$ such that

$$0 \le cg(n) < f(n)$$

- Asymptotic strict lower bound

  Lower bound that is not asymptotically tight

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

(Ex) $an^2 + bn + c = \omega(n)$

$an + b \neq \omega(n)$

Comparisons of Functions

- Related Properties:
  - *Transitivity:*

    $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n)) \Rightarrow f(n) = \Theta(h(n))$.

    Same for O, Ω, o, and ω.
  - *Reflexivity:*

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

    Same for O and Ω.
  - *Symmetry:*

    $f(n) = \Theta(g(n))$ if and only if $g(n) = \Theta(f(n))$.
  - *Transpose symmetry:*

    $f(n) = O(g(n))$ if and only if $g(n) = \Omega(f(n))$.

    $f(n) = \omega(g(n))$ if and only if $g(n) = \omega(f(n))$.

Standard notations and common functions

- Monotonicity:
  - $f(n)$ is *monotonically increasing* if $m \leq n \Rightarrow f(m) \leq f(n)$.
  - $f(n)$ is *monotonically decreasing* if $m \leq n \Rightarrow f(m) \geq f(n)$.
  - $f(n)$ is *strictly increasing* if $m < n \Rightarrow f(m) < f(n)$.
  - $f(n)$ is *strictly decreasing* if $m < n \Rightarrow f(m) > f(n)$.
- Floor and Ceilings: $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x+1$
- Modular arithmetic: $a \bmod n = a - \lfloor a/n \rfloor n$

Polynomials, Exponentials and Logarithms

- Polynomials
  - $a_d n^d + a_{d-1} n^{d-1} + \cdots + a_1 n + a_o$
  - $f(n) = O(n^k)$ for some $k \Rightarrow f(n)$ is polynomially bounded

- Exponentials
  - $a^n \quad (a > 1)$
  - exponential 함수는 polynomial 함수에 비하여 급격히 증가

$$a^n >> n^b \rightarrow \lim_{n->\infty} \frac{n^b}{a^n} = 0 \rightarrow n^b = o(a^n)$$

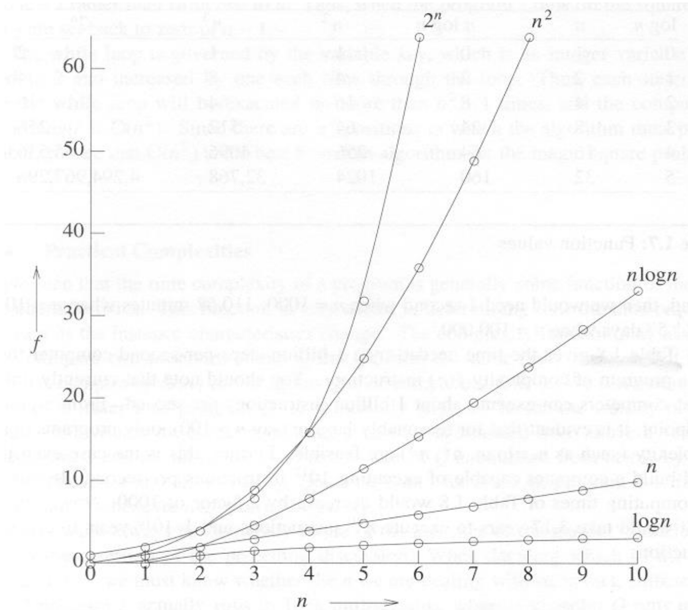
  - exponentially bounded => 무한대의 running time 을 갖는 알고리즘

- Logarithms
  - $\lg n = \log_2 n \quad , \quad \ln n = \log_e n$
  - polynomial 함수는 logarithm 함수에 비하여 급격히 증가

$$n^a >> \lg^b n \rightarrow \lim_{n->\infty} \frac{\lg^b n}{n^a} = 0 \rightarrow \lg^b n = o(n^a)$$

  - $f(n) = O(\lg^k n)$ for some $k \Rightarrow f(n)$ is poly-logarithmically bounded

Polynomials, Exponentials and Logarithms



	Time for $f(n)$ instructions on a 10^9 instr/sec computer						
n	n	$n\log_2 n$	n^2	n^3	n^4	n^{10}	2^n
10	.01μs	.03μs	.1μs	1μs	10μs	10s	1μs
20	.02μs	.09μs	.4μs	8μs	160μs	2.84h	1ms
30	.03μs	.15μs	.9μs	27μs	810μs	6.83d	1s
40	.04μs	.21μs	1.6μs	64μs	2.56ms	121d	18min
50	.05μs	.28μs	2.5μs	125μs	6.25ms	3.1y	13d
100	.10μs	.66μs	10μs	1ms	100ms	3171y	$4*10^{13}$y
1,000	1μs	9.96μs	1ms	1s	16.67min	$3.17*10^{13}$y	$32*10^{283}$y
10,000	10μs	130.03μs	100ms	16.67min	115.7d	$3.17*10^{23}$y	
100,000	100μs	1.66ms	10s	11.57d	3171y	$3.17*10^{33}$y	
1,000,000	1ms	19.92ms	16.67min	31.71y	$3.17*10^{7}$y	$3.17*10^{43}$y	