Growth of Functions
&
Asymptotic Notations

- Algorithm running time
  - Exact running time can be computed for small problem size (small n)
  - Asymptotic performance can be found for large n where *multiplication constants* and *lower-order terms* of the exact running time are dominated.
- Asymptotic performance of algorithm

  How the running time of an algorithm increases with the size of the input in the limit, as the size of the input increases without bound

Asymptotic Notation

Θ-notation

For function $g(n)$, we define $\Theta(g(n))$, big-Theta of n, as:

$\boldsymbol{\Theta(g(n)) = \{ f(n) :}$
$\exists$ positive constants c_1, c_2, and n_0, such that
$\boldsymbol{0 \le c_1 g(n) \le f(n) \le c_2 g(n) \ \forall n \ge n0 \}}$

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

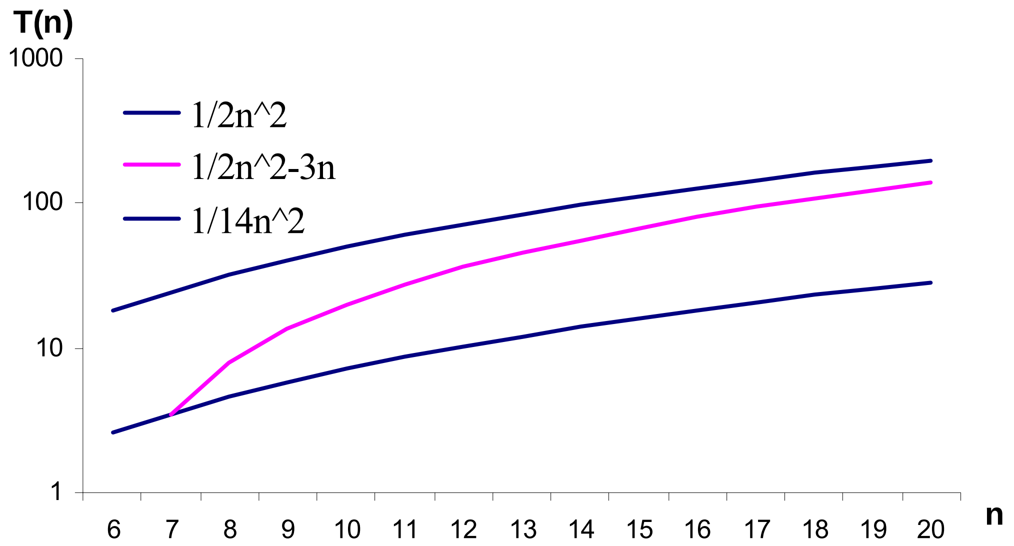
Example

$1/2n^2\text{-}3n=\Theta(n^2)$???

c_1, c_2, n_0 ???

Θ-Notation

$f(\mathrm{n}) = 1/2n^2 - 3n$ $\qquad$ $\mathrm{g}(n) = n^2$



$f(n) = O(g(n))$ & $\quad$ $f(n) = \Omega(g(n)) \rightarrow$ $\quad$ $f(n) = \Theta(g(n))$

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

O-notation

For function $g(n)$, we define $O(g(n))$, big-Oh of n, as:

$O(g(n)) = \{ f(n) :$
$\exists$ **positive constants** c **and** $n_{0,}$ **such that**
$0 \leq f(n) \leq c\, g(n)\ \forall n \geq n0 \}$

$g(n)$ **is an** ***asymptotically upper bound*** **on** $f(n)$**.**
(Worst Case)

Example

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

c, n_0 ???

Ω-notation

For function $g(n)$, we define $\Omega(g(n))$, big-Omega of n, as:

$\Omega(g(n)) = \{ f(n)$:
$\exists$ **positive constants** c **and** $n_{0,}$ **such that**
$0 \leq c\, g(n) \leq f(n) \quad \forall n \geq n0 \}$

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

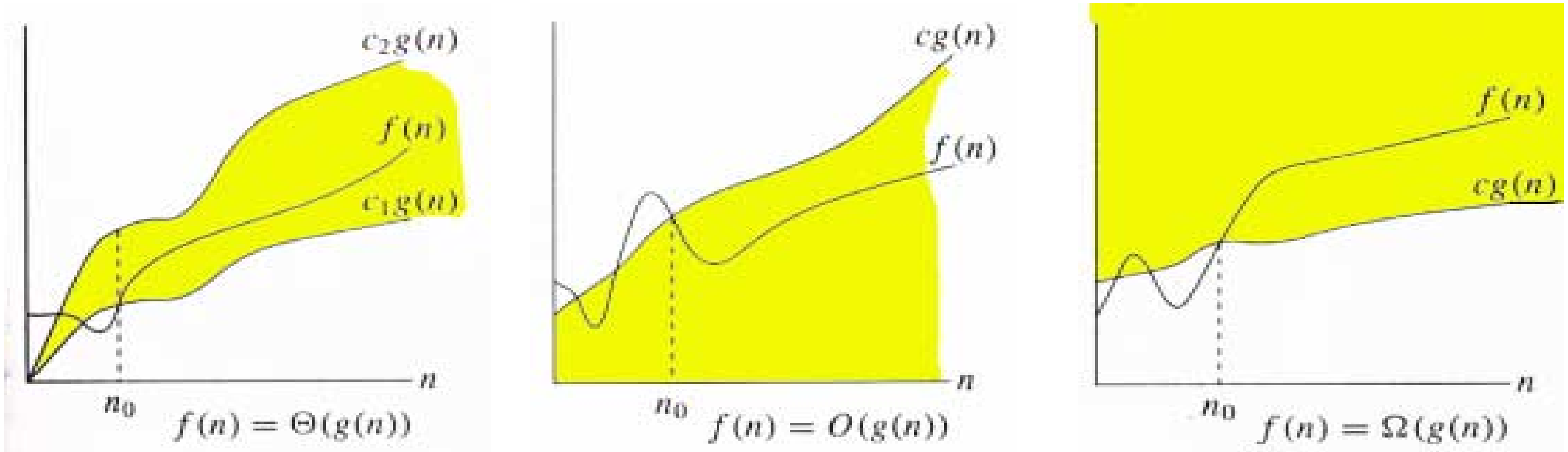
(Best Case)

Example

$\sqrt{n} = \Omega(\lg n)$???

c, n_0 ???

Relations between Θ, Ω, O



Theorem :

For any two functions $g(n)$ and $f(n)$,

$$f(n) = \Theta(g(n)) \text{ iff}$$

$$f(n) = O(g(n)) \text{ and } f(n) = \Omega(g(n)).$$

Asymptotic notations in equation and inequalities

- $2n^2+3n+1 \;=\; 2n^2+\Theta(n)$
  - ✓ there is some function $f(n) = \Theta(n)$ such that $2n^2+3n+1 = 2n^2 + f(n)\ \forall\ n$
- $2n^2+\Theta(n) \;=\; \Theta(n^2)$
  - ✓ for any function $f(n) = \Theta(n)$ there is some function $g(n)= \Theta(n^2)$ such that $2n^2 + f(n) = g(n)\ \forall\ n$.

o-notation

For function $g(n)$, we define $o(g(n))$, little-oh of n, as:

$o(g(n)) = \{ f(n) :$
for any positive constants $c > 0$ $\exists$ a constant
$n_0 > 0$ such that
$0 \le f(n) \le cg(n)$ $\forall n \ge n0$ $\}$

$g(n)$ is not an *asymptotically tight upper bound* for $f(n)$.

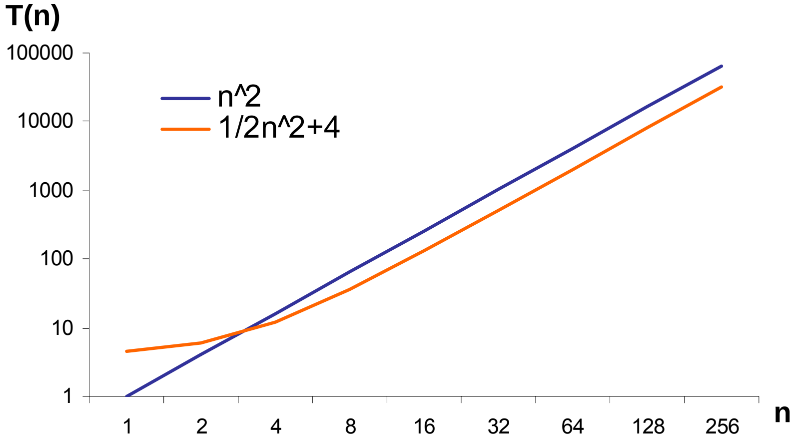
$f(n)$ becomes insignificant relative to $g(n)$ as n approaches infinity:

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

O & o Notations

$f1(n) = ½\, n^2+4$

$g(n) = n^2$

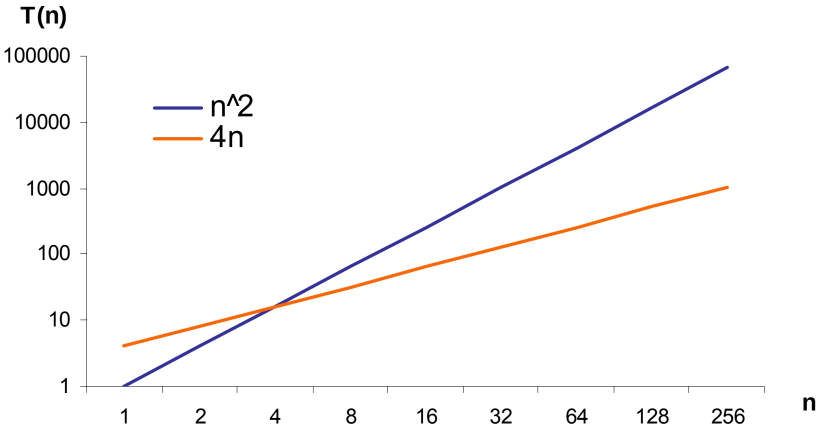


$$f1(n) = O(g(n))$$

***g*(*n*) is *asymptotically tight upper bound* for *f*(*n*)**

$f2(n) = 4n$

$g(n) = n^2$



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

$$f2(n) = o(g(n))$$

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

ω-notation

For function $g(n)$, we define $\omega(g(n))$, little-omega of n, as:

$\boldsymbol{\omega(g(n))} = \{$ **$f(n)$:**
for any positive constants $c > 0$ $\exists$ a constant $n_0 > 0$ such that
$0 \le cg(n) \le f(n) \quad \forall n \ge n0 \}$

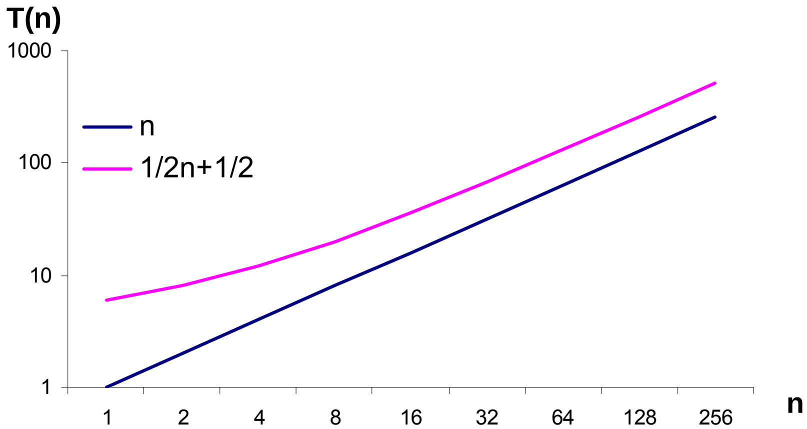
***g*(*n*) is not an *asymptotically tight lower bound* for *f*(*n*).**

$f(n)$ becomes arbitrarily large relative to $g(n)$ as n approaches infinity:

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

Ω & ω Notations

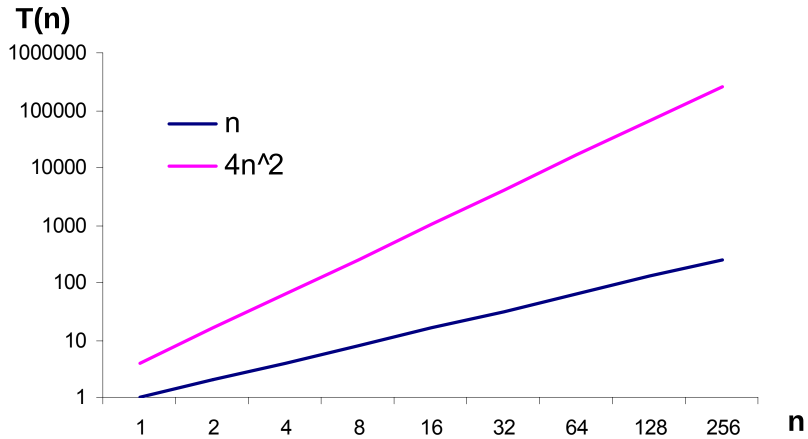
$$f1(n) = 2n + 4$$
$$g(n) = n$$



$$f1(n) = \Omega(g(n))$$

***g*(*n*) is *asymptotically tight lower bound* for *f*(*n*)**

$$f2(n) = 4n^2$$
$$g(n) = n$$



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

$$f2(n) = \omega(g(n))$$

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

Comparison of Functions

- **Transitivity**

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

- **Reflexivity**

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

- **Symmetry**

$$f(n) = \Theta(g(n)) \text{ iff } g(n) = \Theta(f(n))$$

- **Complementarity**

$$f(n) = O(g(n)) \text{ iff } g(n) = \Omega(f(n))$$

$$f(n) = o(g(n)) \text{ iff } g(n) = \omega((f(n))$$

Notes on Functions

$$p(n) = \sum_{i=0}^{d} a_i n^i \qquad a_d \neq 0 \qquad \Rightarrow p(n) = \Theta(n^d)$$

$$\lim_{n \to \infty} \frac{n^b}{a^n} = 0 \qquad \Rightarrow \qquad n^b = o(a^n)$$

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

$$\Rightarrow \qquad n^b = \omega(\lg^a n)$$

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

Examples