Asymptotic Notation

Analysis of Algorithms

An *algorithm* is a finite set of precise instructions for performing a computation or for solving a problem.

- What is the goal of analysis of algorithms?
  - To compare algorithms mainly in terms of running time but also in terms of other factors (e.g., memory requirements, programmer's effort etc.)
- What do we mean by running time analysis?
  - Determine how running time increases as the size of the problem increases.

Types of Analysis

- **Worst case**
  - Provides an upper bound on running time
  - An absolute guarantee that the algorithm would not run longer, no matter what the inputs are
- **Best case**
  - Provides a lower bound on running time
  - Input is the one for which the algorithm runs the fastest
- **Average case**
  - Provides a prediction about the running time
  - Assumes that the input is random

Asymptotic Analysis

- To compare two algorithms with running times $f(n)$ and $g(n)$,

  we need a rough measure that characterizes how fast each function grows.
- Express running time as a function of the input size n (i.e., $f(n)$).
- Compare different functions corresponding to running times.
- Such an analysis is independent of machine time, programming style,etc.
- Compare functions in the limit, that is, **asymptotically!**

  (i.e., for large values of n)

Asymptotic Notation

A way to describe the behavior of functions *in the limit* or *without bounds*.

- The notations are defined in terms of functions whose domains are the set of natural numbers $N=\{0,1,2,\ldots\}$.
- Such notations are convenient for describing the worst-case running time function $T(n)$.
- It can also be extended to the domain of real numbers.

Asymptotic Notation

Example : -

x is asymptotic with x + 1

limit - $\lim_{x \to \infty} f(x) = k$

Roughly translated might read as:

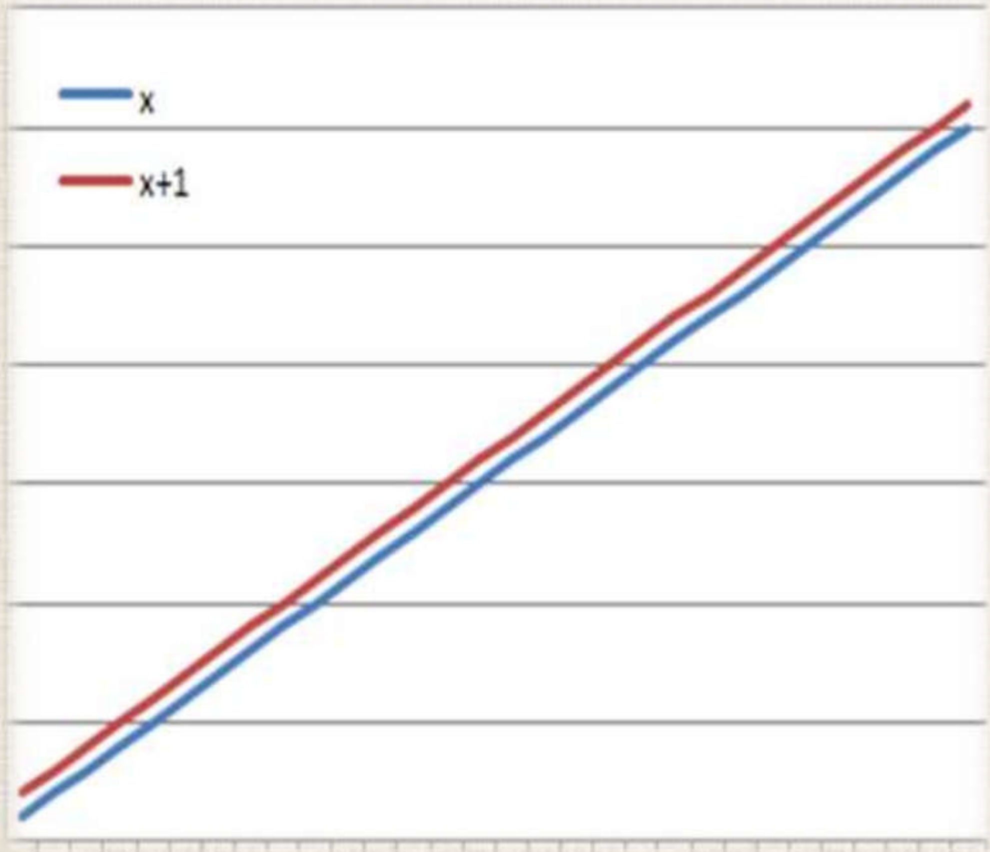
x approaches ∞, f(x) approaches k for

x close to ∞, f(x) is close to k

Two limits often used in analysis are:

$\lim_{x \to \infty} f(x) = \lim_{x \to \infty} f(x) 1/x = 0$

$\lim_{x \to \infty} f(x) = \lim_{x \to \infty} f(x) cx = \infty$ for c>0



Outline

Asymptotic growth rate : -

- Big Oh (O) -notation
- Omega (Ω) -notation
- Theta (Θ) -notation
- Little Oh (o) -notation
- ω-notation

O notation : asymptotic "less than" : $f(n) \le cg(n)$

Ω notation : asymptotic "greater than" : $f(n) \ge c\,g(n)$

Θ notation : asymptotic "equality" : $c_1\,g(n) \le f(n) \le c_2g(n)$

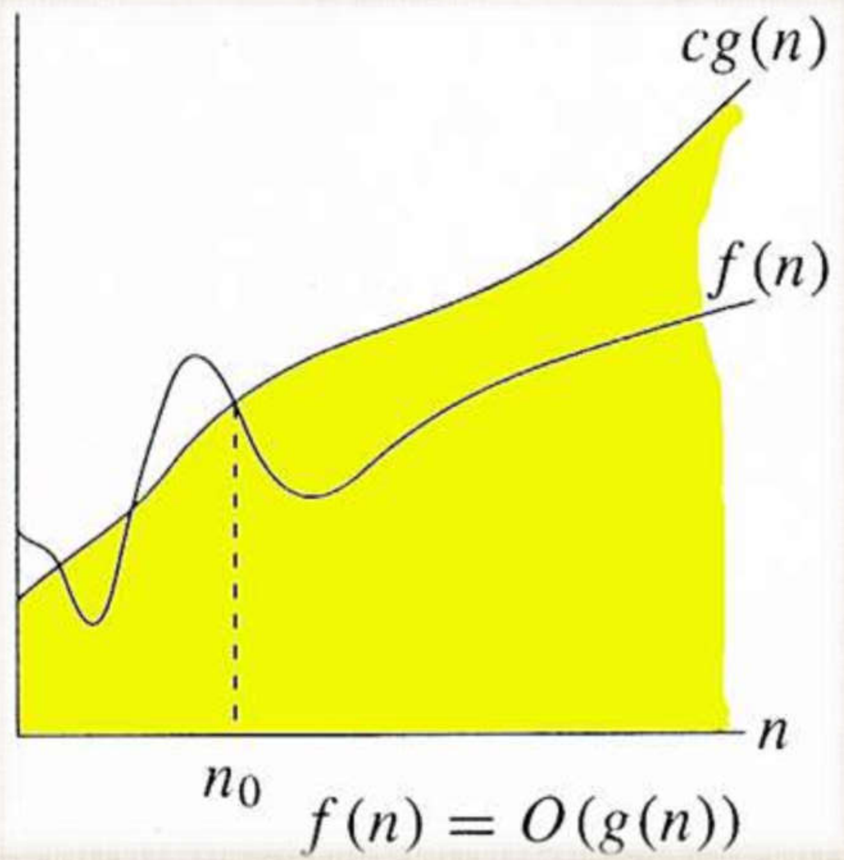
Big-O Notation (Omicron)

possibly *asymptotically* tight **upper** bound for f(n) - Cannot do worse, can do better

- n is the problem size.
- f(n) ∈ O(g(n)) where:

O(g(n)) = { f(n): ∃ positive constants *c*, n_0 such that 0 ≤ f(n) ≤ *c*g(n), ∀ n ≥ n_0}

Meaning for all values of *n* ≥ n_0 f(n) is on or below g(n).



- O(g(n)) is a set of all the functions f(n) that are less than or equal to cg(n), ∀ n ≥ n_0.

If f(n) ≤ cg(n), c > 0, ∀ n ≥ n_0 then f(n) ∈ O(g(n))

Example of Big O notation

Show $2n^2 = O(n^3)$ $0 \le f(n) \le cg(n)$ Defination of $O(g(n))$

<u>Solution</u> :-

$0 \le 2n^2 \le cn^3$ Substitute

$0/n^3 \le 2n^2/n^3 \le cn^3/n^3$ Divide by n^3

Determine C

$0 \le 2/n \le c$

$\lim_{n\to\infty} 2/n = 0$

$2/n$ maximum when n=1

$0 \le 2/1 \le c = 2$ Satisfied by c=2

Determine n_0

$0 \le 2/n_0 \le 2$

$0 \le 2/2 \le n_0$ Satisfied by n_0=1

$0 \le 1 \le n_0 = 1$

$0 \le 2n^2 \le 2n^3$ $\forall\, n \ge n_0$=1

$1000n^2 + 50n = O(n^2)$

with c=1050 and n_0=1

If $f(n) \le cg(n)$, $c > 0$, $\forall\, n \ge n_0$ then $f(n) \in O(g(n))$

Big Omega Notation (Ω)

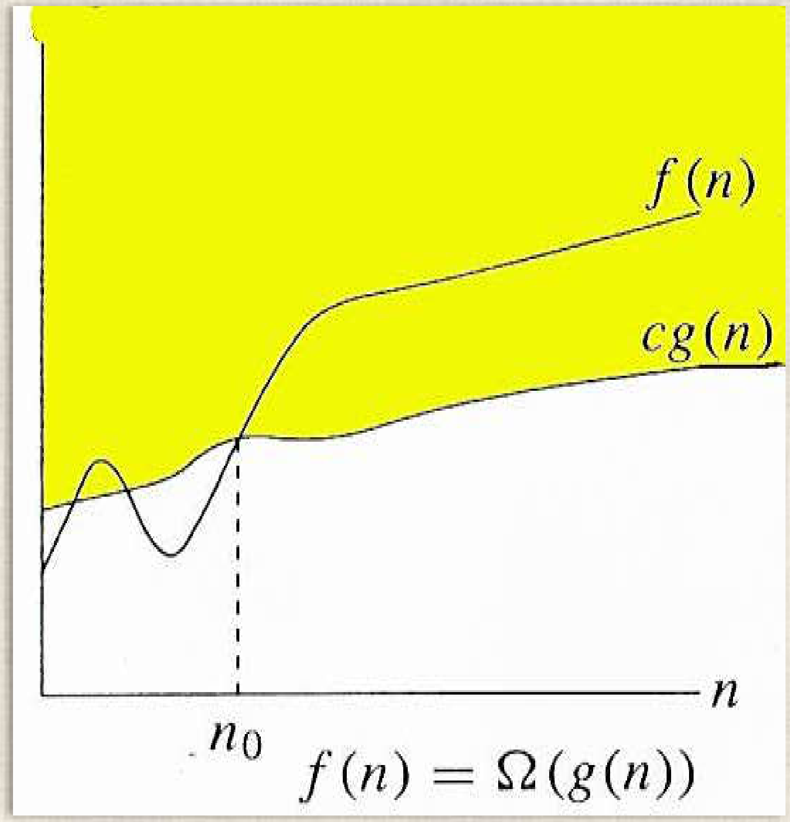
possibly asymptotically tight **lower** bound for f(n) - Cannot do better, can do worse f(n) ∈ Ω(g(n)) where:

Ω(g(n)) = {f(n): ∃ positive constants c > 0, n_0 such that 0 ≤ cg(n) ≤ f(n), ∀ n ≥n_0}

Meaning for all values of n ≥ n_0f(n) is on or above g(n).

Ω(g(n)) is a set of all the functions f(n) that are greater than or equal cg(n), ∀ n ≥n_0.

If cg(n) ≤ f(n), c > 0 and ∀ n ≥n_0, then f(n) ∈ Ω(g(n))



Example of Ω notation

Showthat $3n^2 + n = \Omega(n^2)$

<u>Solution</u> :-

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

$0 \le c\,n^2 \le 3\,n^2 + n$

$0/n^2 \le c\,n^2/n^2 \le 3\,n^2/n^2 + n/n^2$

$0 \le c \le 3 + 1/n$

$\log_{n\to\infty} 3 + 1/n = 3$

$0 \le c \le 3$

c = 3

$3n^2 + n = \Omega(n^2)$ with c=1 and n_0 =1

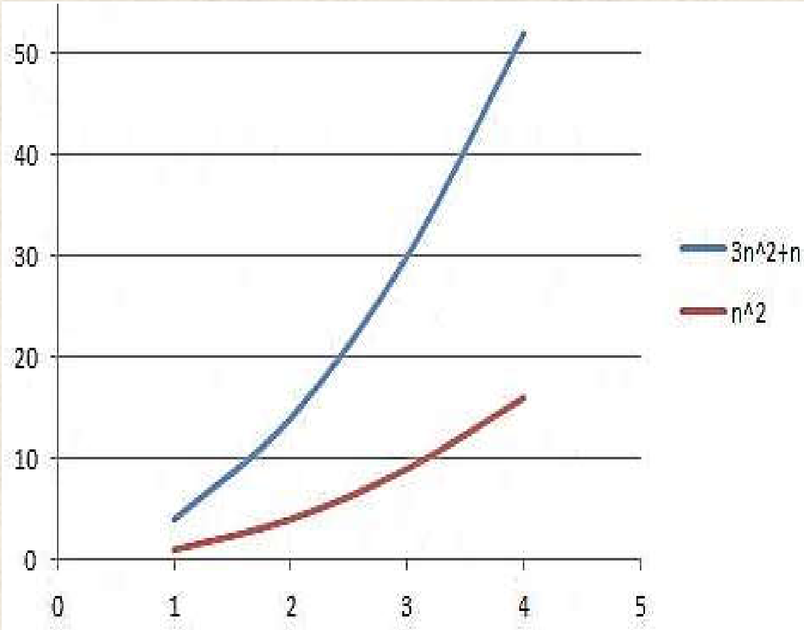
$0 \le 3 \le 3 + 1/n_0$

$-3 \le 3-3 \le 3-3 + 1/n_0$

n_0 =1 satisfies

$-3 \le 0 \le 1/n_0$

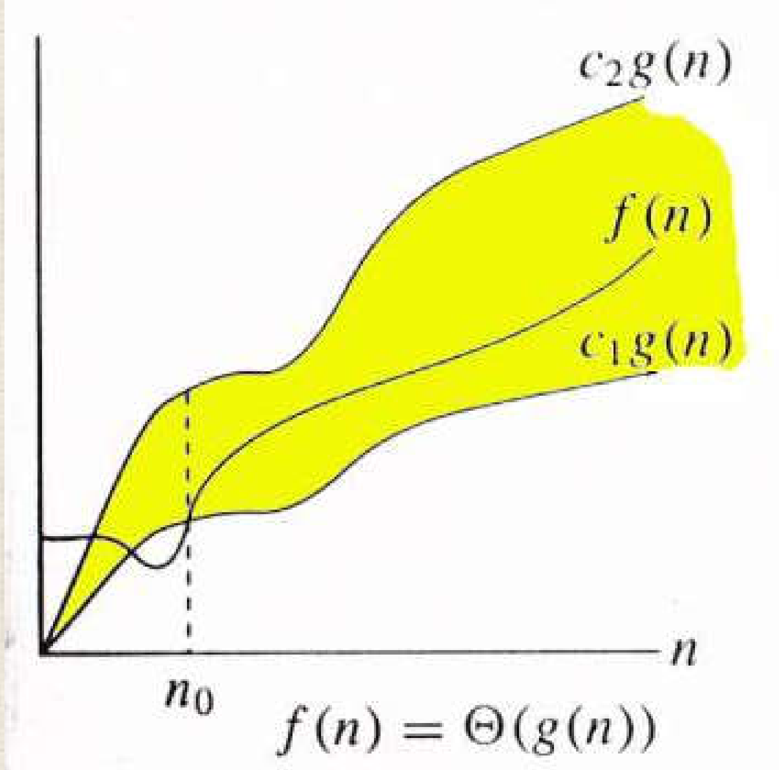
$\log_{n\to\infty} 1/n$



Big Theta Notation(Θ)

asymptotically tight bound for f(n)
$f(n) \in \Theta(g(n))$ where :

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



- Positive means greater than 0.
- $\Theta(g(n))$ is a set of all the functions f(n) that are between c_1 g(n) and c_2g(n), $\forall n \ge n_0$.
- If f(n) is between c_1 g(n) and c_2g(n), $\forall n \ge n_0$, then $f(n) \in \Theta(g(n))$

Example of Θ notation

Solution :-

Show that $\frac{1}{2}n^2 - 3n \in \Theta(n^2)$ Determine $\exists$ c_1, c_2, n_0 *positive constants* such that:

$c_1 n2 \le \frac{1}{2}n^2 - 3n \le c_2 n^2$

$c_1 \le \frac{1}{2} - 3/n \le c_2$ Divide by n^2

O: Determine $c_2 = \frac{1}{2}$

$\frac{1}{2} - 3/n \le c_2$

as $n \rightarrow \infty$, $3/n \rightarrow 0$

$\lim_{n\to\infty} 1/n - 3/n = \frac{1}{2}$

therefore $\frac{1}{2} \le c_2$ o r $c_2 = \frac{1}{2}$ maximum for as $n \rightarrow \infty$ Ω: Determine $c_1 = 1/14$

$0 < c_1 \le \frac{1}{2} - 3/n$

$- 3/n > 0$ minimum for $n_0 = 7$

$0 < c_1 = \frac{1}{2} - 3/7 = 7/14 - 6/14 = 1/14$

n_0: Determine $n_0 = 7$

$c_1 \le \frac{1}{2} - 3/n_0 \le c_2$

$1/14 \le \frac{1}{2} - 3/n_0 \le \frac{1}{2}$

$-6/14 \le -3/n_0 \le 0$

$-n_0 \le -3*14/6 \le 0$

$n_0 \ge 42/6 \ge 0$

$n_0 \ge 7$

Θ: $\frac{1}{2}n^2 - 3n \in \Theta(n^2)$ when $c_1 = 1/14$, $c_2 = \frac{1}{2}$ and $n_0 = 7$

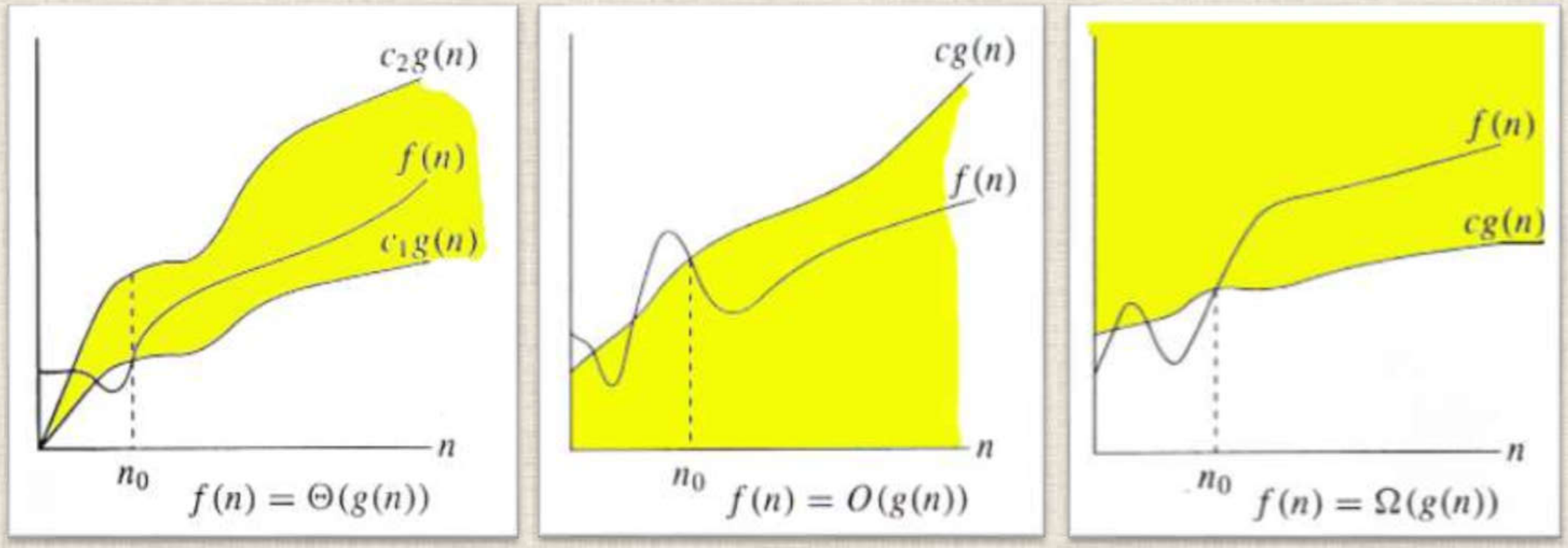
Relations Between Θ, Ω, O

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

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

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

- I.e., $\Theta(g(n)) = O(g(n)) \cap \Omega W(g(n))$
- **In practice, asymptotically tight bounds are obtained from asymptotic upper and lower bounds.**



Little-o Notation (omicron)

non-asymptotically tight ***upper*** *bound for f(n)*

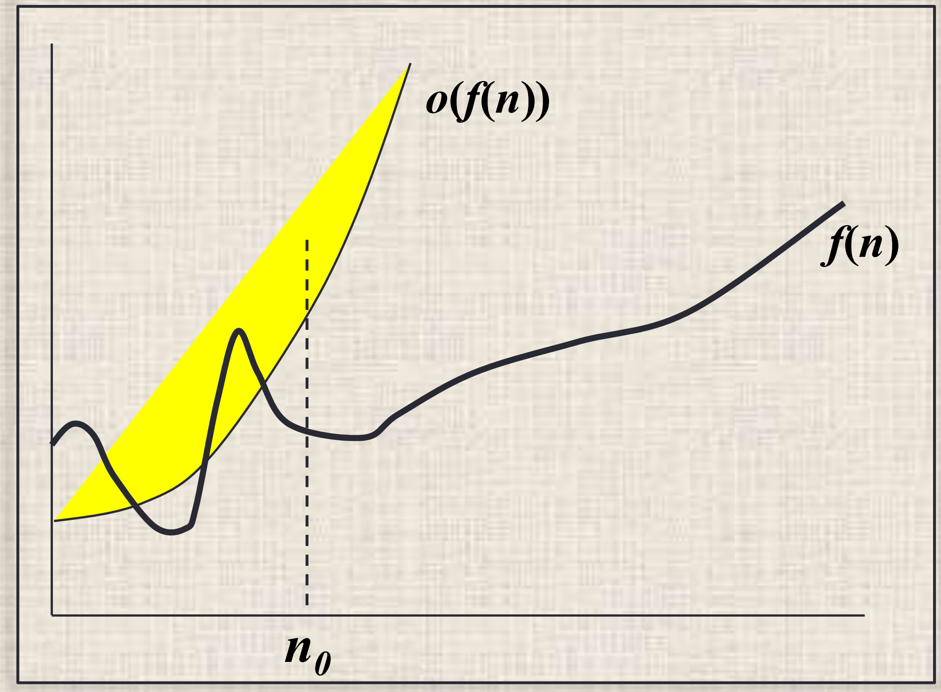
f(n) ∈ o(g(n)) where :

o(g(n)) = {f(n): for **any** constant c > 0, ∃ a constant n_0 >0,
such that: 0 ≤ f(n) **<**cg(n), ∀ n ≥n_0}

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

for **any** 0 < c < ∞

o(g(n)) = {f(n): for **any** constant c > 0, ∃ a constant n_0 >0, such that: 0 ≤ f(n) **<**cg(n), ∀ n ≥n_0}



Little-ω Notation (omega)

*non-asymptotically tight **lower** bound for f(n)*

$f(n) \in \omega(g(n))$ where :

$\omega(g(n))$ = {f(n): for **any** constant $c > 0$, ∃ a constant $n_0 > 0$, such that $0 \le cg(n) < f(n)$, $\forall\, n \ge n_0$}

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

for **any** $0 < c \le \infty$

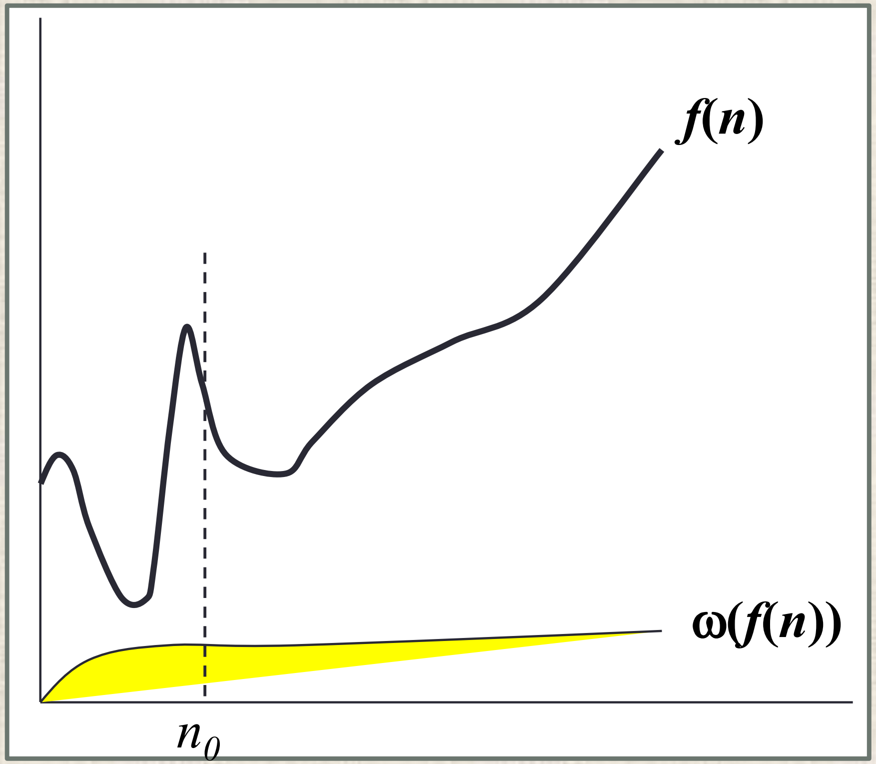
$\Omega(g(n))$ = {f(n): for **some** constant $c > 0$, ∃ a constant $n_0 > 0$, such that $0 \le cg(n) \le f(n)$, $\forall\, n \ge n_0$}

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

for **some** $0 < c \le \infty$

Ω *possibly asymptotically tight **lower** bound*

ω *non-asymptotically tight **lower** bound*



Comparison of Functions

- An imprecise analogy between the asymptotic comparison of two function f and g and the relation between their values:

$$f(n) = O(g(n)) \approx f(n) \leq g(n)$$

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

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

$$f(n) = o(g(n)) \approx f(n) < g(n)$$

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

Limits

- $\lim_{n\to\infty} [f(n) / g(n)] = 0 \Rightarrow f(n) \in o(g(n))$
- $\lim_{n\to\infty} [f(n) / g(n)] < \infty \Rightarrow f(n) \in O(g(n))$
- $0 < \lim_{n\to\infty} [f(n) / g(n)] < \infty \Rightarrow f(n) \in \Theta(g(n))$
- $0 < \lim_{n\to\infty} [f(n) / g(n)] \Rightarrow f(n) \in \Omega(g(n))$
- $\mathit{Lim}_{n\to\infty} [f(n) / g(n)] = \infty \Rightarrow f(n) \in \omega(g(n))$
- $\lim_{n\to\infty} [f(n) / g(n)]$ undefined $\Rightarrow$ can't say

Properties

- **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**

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

- **Complementarity**

  $f(n) = O(g(n))$ *iff* $g(n) = \Omega(f(n))$

  $f(n) = o(g(n))$ *iff* $g(n) = \omega((f(n))$

Asymptotic notation in equations and inequalities

- On the right-hand side :-

  $\Theta(n^2)$ stands for some anonymous function in $\Theta(n^2)$

  $2n^2 + 3n + 1 = 2n^2 + \Theta(n)$ means:

  There exists a function $f(n) \in \Theta(n)$ such that

  $$2n^2 + 3n + 1 = 2n^2 + f(n)$$

- On the left-hand side :-

  $2n^2 + \Theta(n) = \Theta(n^2)$ means :

  No matter how the anonymous function is chosen on the left-hand side, there is a way to choose the anonymous function on the right-hand side to make the equation valid.

Basic Asymptotic Efficiency Classes

Class	Name	Comment
1	***constant***	– Instructions are executed once or a few times
log n	***logarithmic***	– A big problem is solved by cutting the original problem in smaller sizes, by a constant fraction at each step
n	***linear***	– A small amount of processing is done on each input element
n log n	***linearithmic***	– A problem is solved by dividing it into smaller problems, solving them independently and combining the solution
n^2	***quadratic***	– Typical for algorithms that process all pairs of data items (double nested loops)
n^3	***cubic***	– Processing of triples of data (triple nested loops)
2^n	***exponential***	– Few exponential algorithms are appropriate for practical use
n!	***factorial***	– Typical for algorithms that generate all permutations of an *n*-element set.