Asymptotic Analysis

Buffalo once roamed the plains in large numbers.

Asymptotic Complexity

- Running time of an algorithm as a function of input size *n*, **for large *n***.
- Expressed using only the **highest-order term** in the expression for the exact running time.
- Describes behavior of function in the limit.
- Written using ***Asymptotic Notation***.

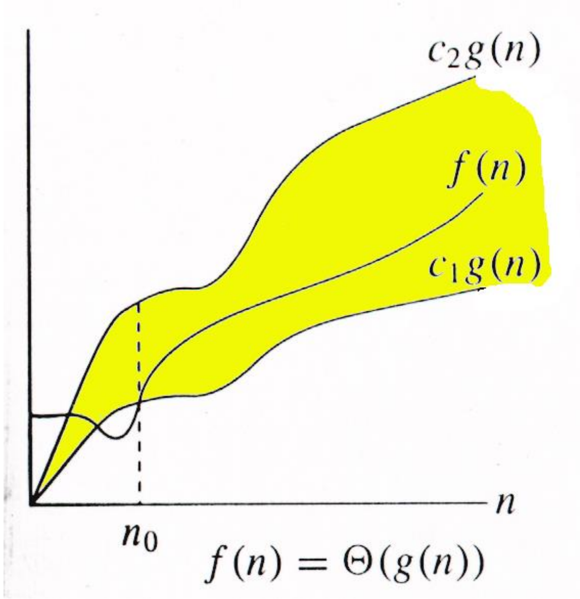
Asymptotic Notation

- **Θ, *O*, Ω, *o*, ω**
- Defined for functions over the natural numbers.
  - **Ex:** $f(n) = \Theta(n^2)$.
    - Describes how $f(n)$ grows in comparison to n^2.
- Defines a ***set*** of functions; in practice used to compare growth rate of two functions.
- The notations describe the relationship between the defining function and a defined set of functions.

Θ-notation

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

$\Theta(g(n))$ = {$f(n)$: $\exists$ positive constants c_1, c_2, and n_0, such that $\forall n \geq n_0$, we have $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$}



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

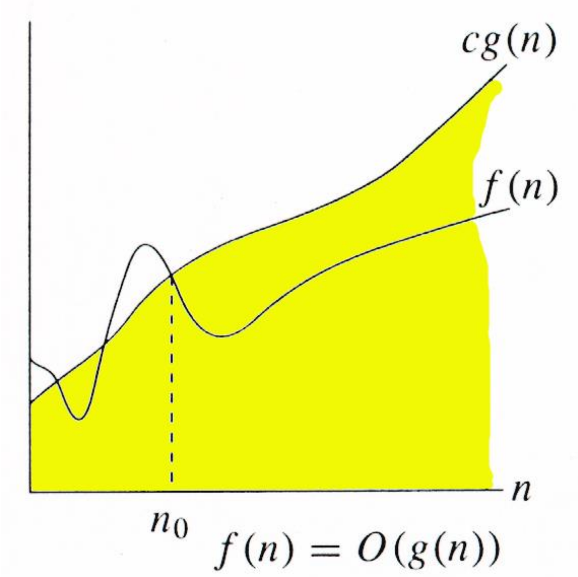
Intuitively: Set of all functions that have the same *rate of growth* as $g(n)$.

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

O-notation

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

$O(g(n))$ = $\{f(n)$:
$\exists$ positive constants c and n_0,
such that $\forall n \geq n_0$, we have $0 \leq f(n) \leq cg(n)$ $\}$



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

Intuitively: Set of all functions whose *rate of growth* is the same as or lower than that of $g(n)$.

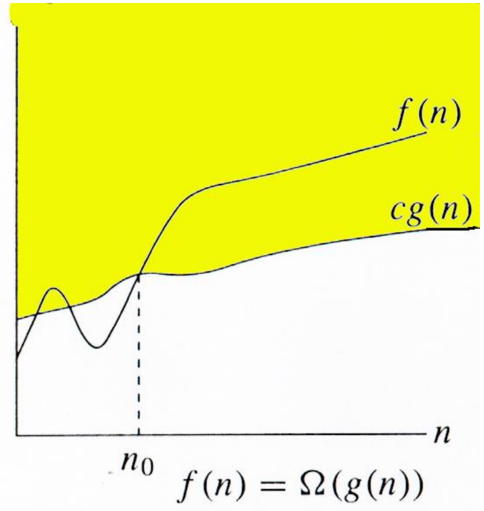
$g(n)$ **is an *asymptotic upper bound* for $f(n)$.**

$$f(n) = \Theta(g(n)) \Rightarrow f(n) = O(g(n))$$
$$\Theta(g(n)) \subset O(g(n))$$

Ω -notation

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

$\Omega(g(n)) = \{f(n) :$
$\exists$ positive constants c and n_0,
such that $\forall n \geq n_0$,
we have $0 \leq cg(n) \leq f(n)\}$

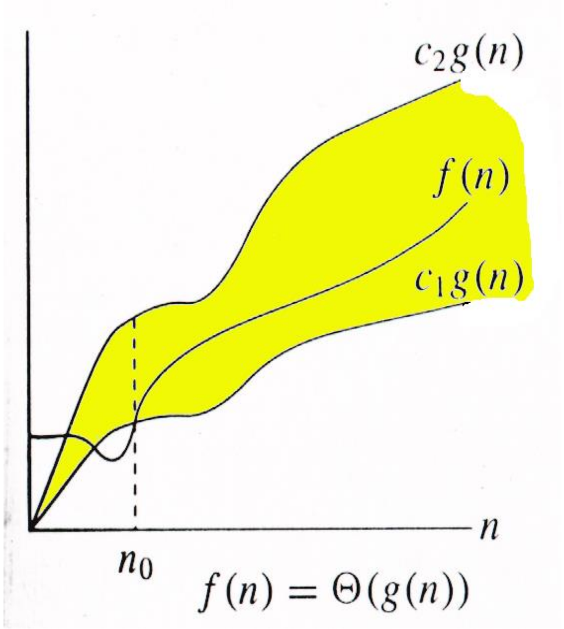


Intuitively: Set of all functions whose *rate of growth* is the same as or higher than that of $g(n)$.

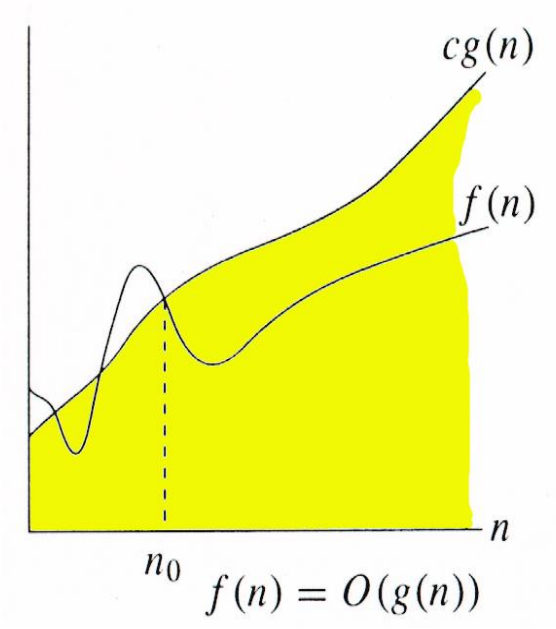
$g(n)$ **is an *asymptotic lower bound* for $f(n)$.**

$$f(n) = \Theta(g(n)) \Rightarrow f(n) = \Omega(g(n))$$
$$\Theta(g(n)) \subset \Omega(g(n))$$

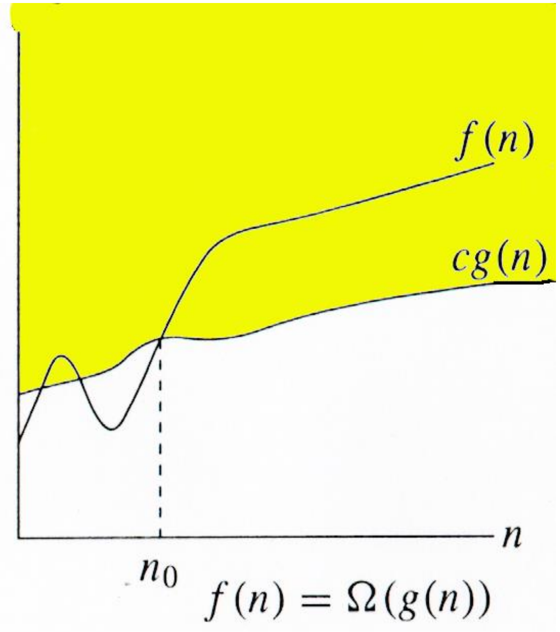
Relationship Between Θ, *O*, Ω



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



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



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

Relationship Between Θ, Ω, *O*

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

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

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

- $\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$

- In practice, asymptotically tight bounds are obtained from asymptotic upper and lower bounds.

o-notation

For a given function $g(n)$, the set little-*o*:

$o(g(n)) = \{f(n): \forall\ c > 0,\ \exists\ n_0 > 0$ such that $\forall\ n \geq n_0$, we have $0 \leq f(n) < cg(n)\}$.

Intuitively: Set of all functions whose *rate of growth* is lower than that of $g(n)$.

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

ω -notation

For a given function $g(n)$, the set little-omega:

$\omega(g(n))$ = $\{f(n)$: $\forall$ $c > 0$, $\exists$ $n_0 > 0$ such that $\forall$ $n \geq n_0$, we have $0 \leq cg(n) < f(n)\}$.

Intuitively: Set of all functions whose *rate of growth* is higher than that of $g(n)$.

$g(n)$ is a ***lower bound*** for $f(n)$ that is not asymptotically tight.

Comparison of Functions

$$f \leftrightarrow g \approx a \leftrightarrow b$$

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

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

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

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

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

Limit Definitions of Asymptotic Notations

- *o*-notation (Little-o):
  - $f(n)$ becomes insignificant relative to $g(n)$ as *n* approaches infinity:

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

- ω -notation (Little-omega):
  - $f(n)$ becomes arbitrarily large relative to $g(n)$ as *n* approaches infinity:

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

Limit Definitions of Asymptotic Notations

- Θ-notation (Big-theta):
  - $f(n)$ relative to $g(n)$ *equals a constant, c, which is greater than 0 and less than infinity* as *n* approaches infinity:

$$0 < \lim_{n \to \infty} [f(n) / g(n)] < \infty$$

Limit Definitions of Asymptotic Notations

- *O*-notation (Big-o):
  - $f(n) \in O(g(n)) \Rightarrow f(n) \in \Theta(g(n))$ and $f(n) \in o(g(n))$
  - $f(n)$ relative to $g(n)$ *equals some value less than infinity* as *n* approaches infinity:

$$\lim_{n \to \infty} [f(n) / g(n)] < \infty$$

- Ω-notation (Big-omega):
  - $f(n) \in \Omega(g(n)) \Rightarrow f(n) \in \Theta(g(n))$ and $f(n) \in \omega(g(n))$
  - $f(n)$ relative to $g(n)$ *equals some value greater than 0* as *n* approaches infinity:

$$0 < \lim_{n \to \infty} [f(n) / g(n)]$$

Examples

Use limit definitions to prove:

- $10n - 3n \in O(n^2)$

- $3n^4 \in \Omega(n^3)$

- $n^2/2 - 3n \in \Theta(n^2)$

- $2^{2n} \in \Theta(2^n)$

Examples

Use limit definitions to prove:

- $10n - 3n \in O(n^2)$

  $\lim_{n\to\infty} [\, 10n - 3n \,/\, n^2 \,] = 0$

- $3n^4 \in \Omega(n^3)$

  $\lim_{n\to\infty} [\, 3n^4 \,/\, n^3 \,] = \infty$

- $n^2/2 - 3n \in \Theta(n^2)$

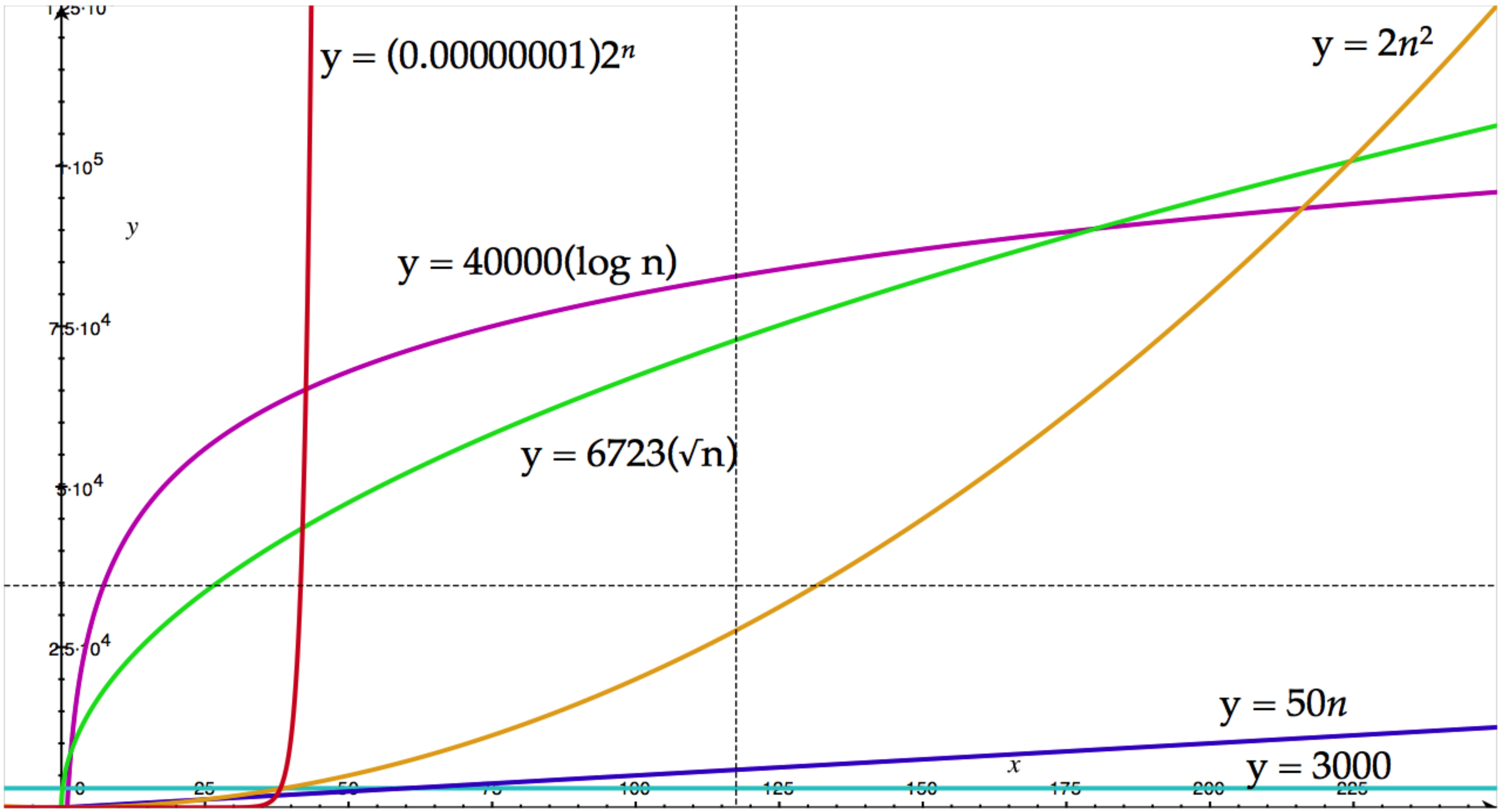
  $\lim_{n\to\infty} [\, n^2/2 - 3n \,/\, n^2 \,] = 1/2$

- $2^{2n} \in \Theta(2^n)$

  $\lim_{n\to\infty} [\, 2^{2n} \,/\, 2^n \,] = 4$

Why Do We Care?

Big-O Examples



Complexity and Tractability

	$T(n)$						
n	n	$n \log n$	n^2	n^3	n^4	n^{10}	2^n
10	.01ms	.03ms	.1ms	1ms	10ms	10s	1ms
20	.02ms	.09ms	.4ms	8ms	160ms	2.84h	1ms
30	.03ms	.15ms	.9ms	27ms	810ms	6.83d	1s
40	.04ms	.21ms	1.6ms	64ms	2.56ms	121d	18m
50	.05ms	.28ms	2.5ms	125ms	6.25ms	3.1y	13d
100	.1ms	.66ms	10ms	1ms	100ms	3171y	4´ 10^{13}y
10^3	1ms	9.96ms	1ms	1s	16.67m	3.17´ 10^{13}y	32´ 10^{283}y
10^4	10ms	130ms	100ms	16.67m	115.7d	3.17´ 10^{23}y	
10^5	100ms	1.66ms	10s	11.57d	3171y	3.17´ 10^{33}y	
10^6	1ms	19.92ms	16.67m	31.71y	3.17´ 10^7y	3.17´ 10^{43}y	

Assuming the microprocessor performs 1 billion ops/s.