COMP 355
Advanced Algorithms

Asymptotics

Rhodes College

Asymptotic Analysis

Asymptotic analysis is based on two simplifying assumptions

- **Large input sizes:** We are most interested in how the running time grows for large values of *n*.
- **Ignore constant factors:** The actual running time of the program depends on various constant factors in the implementation (coding tricks, optimizations in compilation, speed of the underlying hardware, etc). Therefore, we will ignore constant factors.

Rhodes College

Large Input Sizes

n	$T_1(n)$	$T_2(n)$	$T_1(n)/T_2(n)$
10	0.001 sec	0.001 sec	1
100	1 sec	0.01 sec	100
1000	17 min	0.1 sec	10,000
10,000	11.6 days	1 sec	1,000,000

$$T_1(n) = n^3$$
$$T_2(n) = 100n$$

As input sizes grow, the performance of the asymptotically poorer algorithm degrades much more rapidly.

Other Asymptotic Forms

- Ω ("big-omega"),
- Θ ("theta"),
- o ("little-oh"),
- ω ("little-omega")

Notation	Relational Form	Limit Definition
$f(n)$ is $o(g(n))$	$f(n) \prec g(n)$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
$f(n)$ is $O(g(n))$	$f(n) \preceq g(n)$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$ or 0
$f(n)$ is $\Theta(g(n))$	$f(n) \approx g(n)$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$
$f(n)$ is $\Omega(g(n))$	$f(n) \succeq g(n)$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$ or ∞
$f(n)$ is $\omega(g(n))$	$f(n) \succ g(n)$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty.$

Rhodes College

Asymptotic Notation

- *Asymptotic notation* - represents a function by its fastest growing term and ignores constant factors
- Example: $T(n) = 13n^3 + 5n^2 - 17n + 16$
  - As n becomes large, the $13n^3$ term dominates the others.
  - Running Time grows "on the order of" n^3.
  - $T(n) \in \Theta(n^3)$

Rhodes College

Formal Definition of Θ

Definition: Given any function $g(n)$, we define $\Theta(g(n))$ to be a set of functions:

$$\Theta(g(n)) = \{f(n) \mid \text{there exist strictly positive constants } c_1, c_2, \text{ and } n_0 \text{ such that } 0 \le c_1 g(n) \le f(n) \le c_2 g(n) \text{ for all } n \ge n_0\}.$$

- Intuitively, what we want to say with "$f(n) \epsilon \Theta(g(n))$" is that $f(n)$ and $g(n)$ are *asymptotically equivalent*. (same growth rates for large n)
- Example:
  $4n^2$, $(8n^2 + 2n - 3)$, $(n^2/5 + \sqrt{n} - 10 \log n)$, and $n(n - 3)$ are all intuitively *asymptotically equivalent*, since as n becomes large, the dominant (fastest growing) term is some constant times n^2

Example

Consider the function $f(n) = 8n^2 + 2n - 3$.

Keep the largest term and throw away the constants → $f(n) \in \Theta(n^2)$

Need to show that

1. $f(n)$ grows asymptotically at least as fast as n^2
2. $f(n)$ grows no faster asymptotically than n^2

Rhodes College

O-notation and Ω-notation

- O-notation: asymptotic upper bounds

Definition: Given any function $g(n)$,

$O(g(n)) = \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0$

- Ω-notation: asymptotic lower bounds

Definition: Given any function $g(n)$,

$\Omega(g(n)) = \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq cg(n) \leq f(n)$ for all $n \geq n_0$

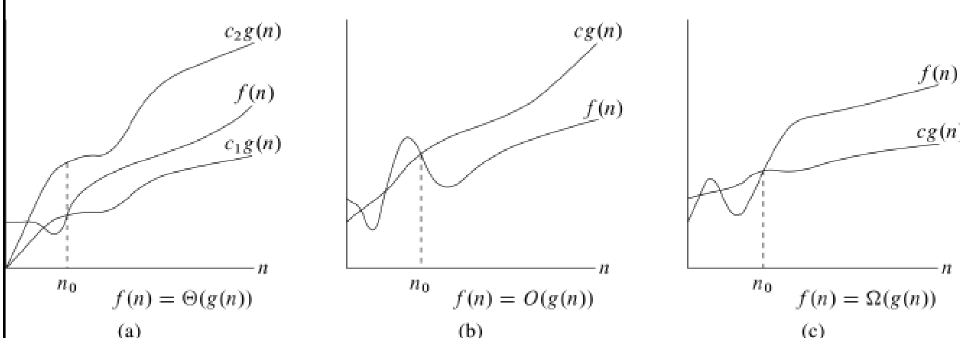
Rhodes College

Example

$$f(n) = 3n^2 + 4\text{n} \;\in\; \Theta(n^2)$$

- $f(n)$ not in $\Theta(n)$ or $\Theta(n^3)$
- $f(n) \in O(n^2)$ $and\ in \in O(n^3)$, not in $\text{O}(n)$
- $f(n) \in \Omega(n^2)$ $and\ in \in \Omega(n)$, not in $\Omega(n^3)$

Graph Depictions



o-notation

- Upper bound provided by O-notation may or may not be asymptotically tight
  - $2n^2 = O(n^2)$ is tight; $2n = O(n^2)$ is not
- Use o-notation (little-o) to denote an upper bound that is not asymptotically tight

Definition:

o($g(n)$) = {$f(n)$ | for any positive constant c > 0, there exists a constant $n \geq n_0$ such that $0 \leq f(n) < cg(n)$ for all $n \geq n_0$}

Example:

$2n = o(n^2)$, but $2n^2 \neq o(n^2)$

Rhodes College

ω-notation

- Lower bound provided by Ω-notation may or may not be asymptotically tight
- Use ω-notation (little-omega) to denote a lower bound that is not asymptotically tight

Definition:

ω($g(n)$) = {$f(n)$ | for any positive constant c > 0, there exists a constant $n_0 \geq 0$ such that $0 \leq cg(n) < f(n)$ for all $n \geq n_0$}

Example:

$n^2/2 = \omega(n)$, but $n^2/2 \neq \omega(n^2)$

Rhodes College

Properties

- Transitivity.
  - If f = O(g) and g = O(h) then f = O(h).
  - If f = Ω(g) and g = Ω(h) then f = Ω(h).
  - If f = Θ(g) and g = Θ(h) then f = Θ(h).
  - If f = o(g) and g = o(h) then f = o(h).
  - If f = ω (g) and g = ω (h) then f = ω(h).
- Additivity.
  - If f = O(h) and g = O(h) then f + g = O(h).
  - If f = Ω(h) and g = Ω(h) then f + g = Ω(h).
  - If f = Θ(h) and g = Θ(h) then f + g = Θ(h).

Rhodes College

L'Hôpital's rule

L'Hôpital's rule: If $f(n)$ and $g(n)$ both approach 0 or both approach ∞ in the limit, then

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

where $f'(n)$ and $g'(n)$ denote the derivatives of f and g relative to n.

Rhodes College

Exponentials and Logarithms

The terminology $\lg^b n$ means $(\lg n)^b$

Lemma: Given any positive constants $a > 1$, b, and c:

$$\lim_{n\to\infty} \frac{n^b}{a^n} = 0 \qquad \lim_{n\to\infty} \frac{\lg^b n}{n^c} = 0.$$

- Polynomials always grow more slowly than exponentials

$$n^{500} \in O(2^n)$$

- Logarithmic powers grow more slowly than any polynomial

$$\lg^{500} n \in O(n)$$

Practice

Work in groups:

In each case, put the two functions in increasing order of asymptotic growth rate. That is, indicate whether f ≺ g (meaning that f(n) is o(g(n))), g ≺ f (meaning that f(n) is ω(g(n))) or f ≈ g (meaning that f(n) is (g(n))).

	f(n)	g(n)
(a)	$10n^3 + n \lg n$	$n^3 + n \lg^2 n$
(b)	2^n	$3^{(n/2)}$
(c)	$\lg(2^n)$	$\lg(3^n)$
(d)	$\lg\sqrt{n}$	$\sqrt{lg\ n}$
(e)	$n^{\lg 4}$	$2^{\lg n}$

Next Time

- Master Theorem