UNIVERSITY OF NORTH TEXAS

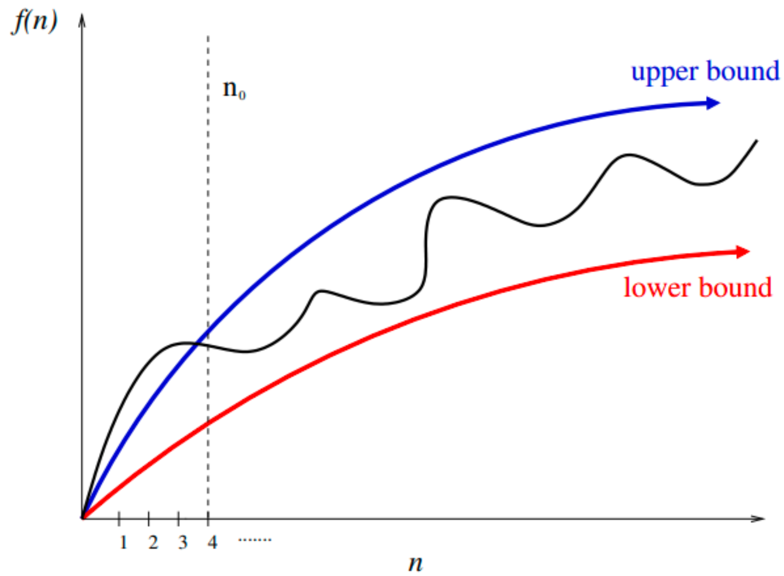
CSCE 3110
Data Structures and Algorithms

Algorithm Analysis I (cont.)

Reading: Weiss, chap. 2

Credits of some slides go to S. Skiena at SUNY Stony Brook.

Exact Analysis is Hard!



- It easier to talk about upper and lower bounds of the function.

Asymptotic notation (O, Ω, Θ) are adopted to deal with complexity functions.

Asymptotic Notation

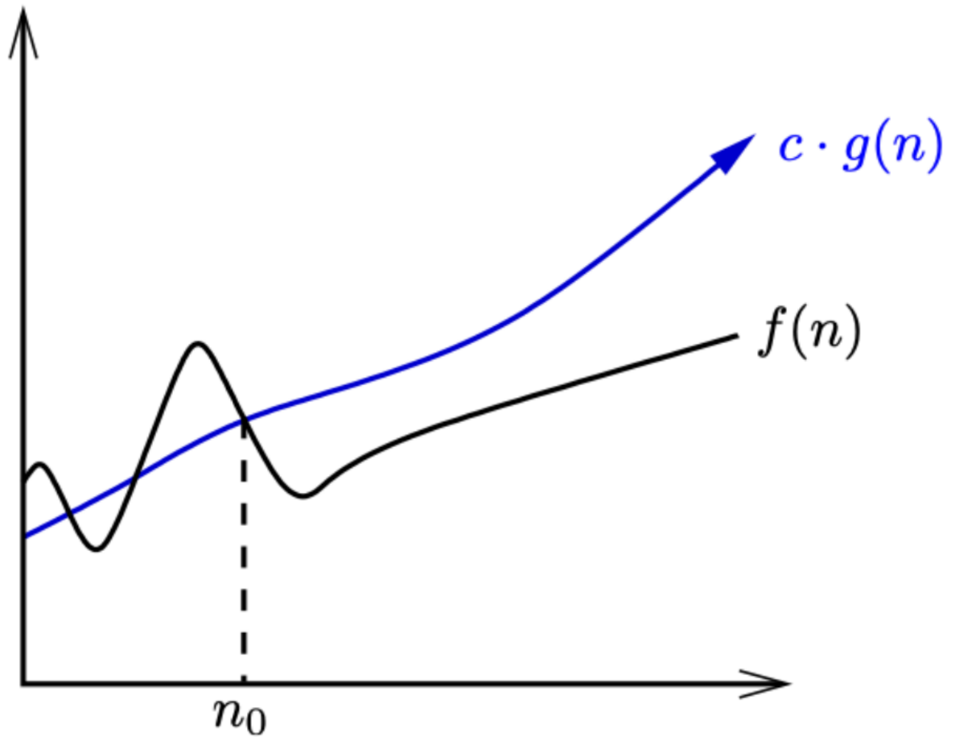
- Asymptotic notation
  - Big Oh
  - Big Omega
  - Big Theta
  - Little Oh
  - Little Omega

Asymptotic Notation

- Asymptotic notation
  - Big Oh
  - Big Omega
  - Big Theta
  - Little Oh
  - Little Omega

Asymptotic Notation - Big Oh, O

- $O \approx \leq$
- Some constant multiple of $g(n)$ is an asymptotic upper bound of $f(n)$, possibly not tight



Asymptotic Notation - Big Oh, O

$$O\big(g(n)\big) = \{f(n)\colon \exists\ c > 0, \exists n_0\ s.t. \forall n \geq n_0 \colon 0 \leq f(n) \leq cg(n)\}$$

- In plain English: $O(g(n))$ are all functions $f(n)$ for which there exists two positive constants c and n_0 such that for all $n \geq n_0$, $0 \leq f(n) \leq cg(n)$
  - $g(n)$ is an asymptotic **upper bound** for $f(n)$
- Intuitively, you can think of O as "$\leq$" for functions
- If $f(n) \in O(g(n))$, we write $f(n) = O(g(n))$

NOTE: The definitions imply a constant n_0 beyond which they are satisfied. We do not care about small values of n.

Asymptotic Notation - Big Oh, O

- Examples

Is $2^{n+1} = O(2^n)$?

Is $2^{2n} = O(2^n)$?

Asymptotic Notation - Big Oh, O

- Examples

$$2^{n+1} = O(2^n)$$

$$2^{2n} \neq O(2^n)$$

Asymptotic Notation - Big Oh, O

- More examples
  - $2n^2 = O(n^3)$

Asymptotic Notation - Big Oh, O

- More examples
  - $2n^2 = O(n^3)$
  - $n = O(n^2)$
  - $\frac{n}{1000} = O(n^2)$
  - $n^{1.999} = O(n^2)$

Asymptotic Notation - Big Oh, O

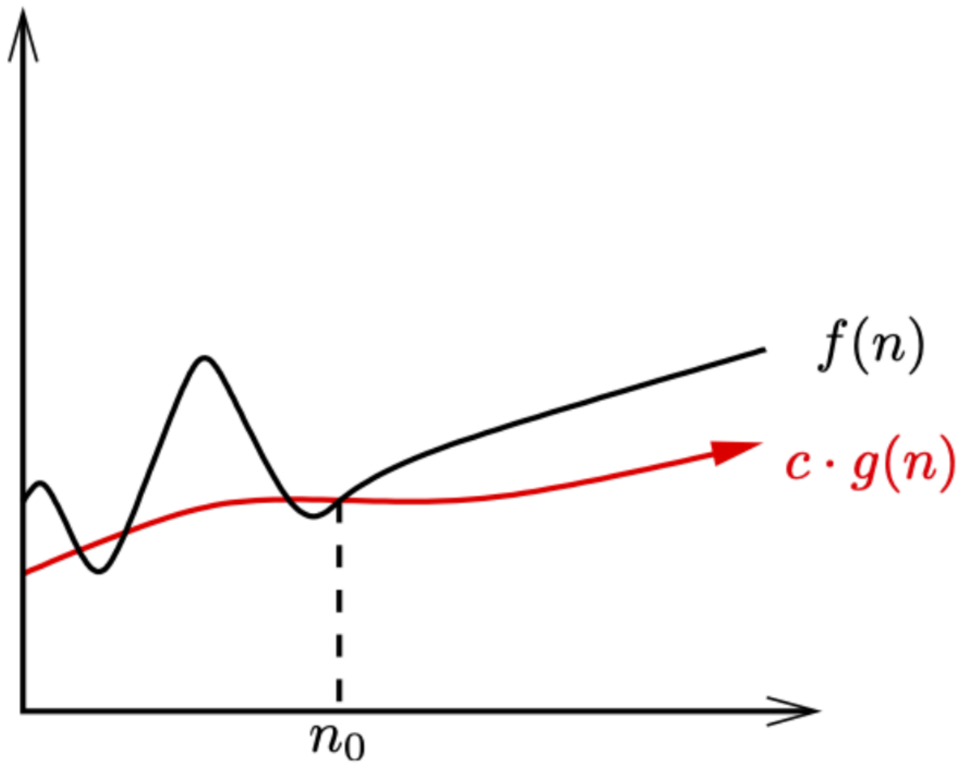
- More examples
  - $2n^2 = O(n^3)$
  - $n = O(n^2)$
  - $\frac{n}{1000} = O(n^2)$
  - $n^{1.999} = O(n^2)$
  - $n^2 + n = O(n^2)$
  - $n^2 + 1000n = O(n^2)$
  - $1000n^2 + 1000n = O(n^2)$

Asymptotic Notation

- Asymptotic notation
  - Big Oh
  - Big Omega
  - Big Theta
  - Little Oh
  - Little Omega

Asymptotic Notation - Big Omega, Ω

- $O \approx \geq$
- Some constant multiple of $g(n)$ is an asymptotic lower bound of $f(n)$, possibly not tight



Asymptotic Notation - Big Omega, Ω

$$\Omega\big(g(n)\big) = \{f(n)\colon \exists\, c > 0, \exists n_0\ s.t.\, \forall n \geq n_0\colon 0 \leq cg(n) \leq f(n)\}$$

- In plain English: $\Omega(g(n))$ are all functions $f(n)$ for which there exists two positive constants c and n_0 such that for all $n \geq n_0$, $0 \leq cg(n) \leq f(n)$
  - $g(n)$ is an asymptotic **lower bound** for $f(n)$
- Intuitively, you can think of Ω as "≥" for functions
- If $f(n) \in \Omega(g(n))$, we write $f(n) = \Omega(g(n))$

NOTE: The definitions imply a constant n_0 beyond which they are satisfied. We do not care about small values of n.

Asymptotic Notation - Big Omega, Ω

- Examples
  - $n^3 = \Omega(n^2)$
  - $n^{2.0001} = \Omega(n^2)$

Asymptotic Notation - Big Omega, Ω

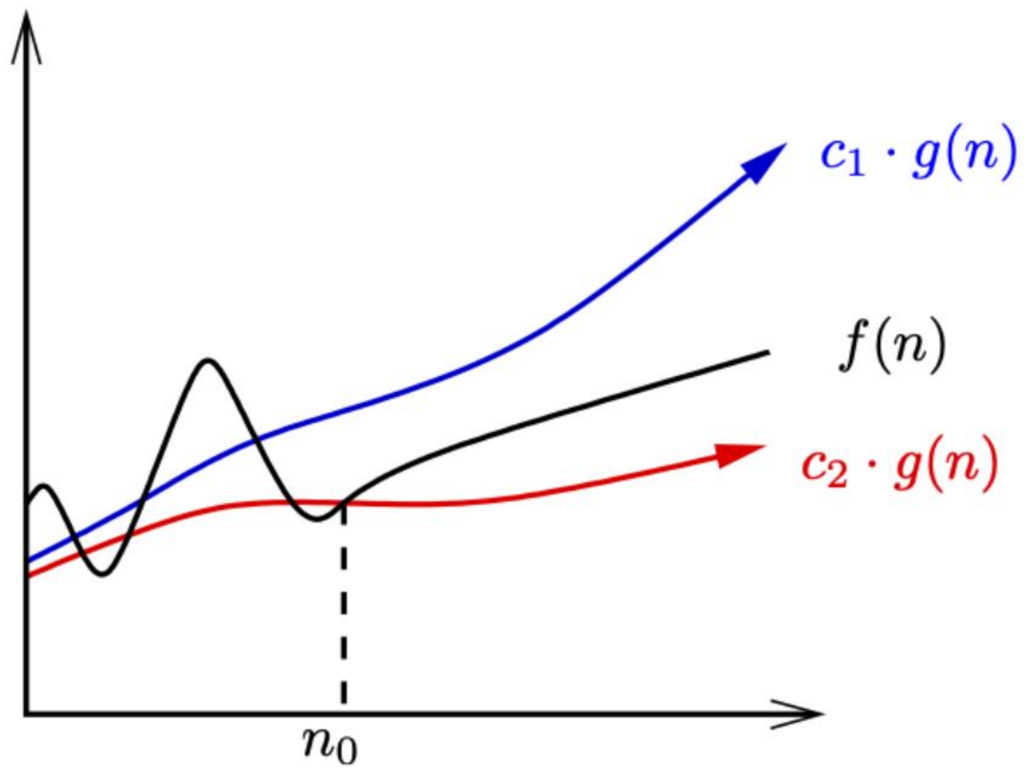
- Examples
  - $n^3 = \Omega(n^2)$
  - $n^{2.0001} = \Omega(n^2)$
  - $n^2 = \Omega(n^2)$
  - $n^2 + n = \Omega(n^2)$
  - $1000n^2 + n = \Omega(n^2)$
  - $1000n^2 + 1000n = \Omega(n^2)$
  - $\sqrt{n} = \Omega(\log n)$

Asymptotic Notation

- Asymptotic notation
  - Big Oh
  - Big Omega
  - Big Theta
  - Little Oh
  - Little Omega

Asymptotic Notation – Theta, Θ

- $\Theta \approx =$
- Some constant multiple of $g(n)$ is an asymptotic tight bound of $f(n)$



Asymptotic Notation – Theta, Θ

$$\Theta(g(n)) = \{f(n)\colon \exists\, c_1 > 0, \exists\, c_2 > 0, \exists n_0\ s.t. \forall n \geq n_0\colon$$
$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)\}$$

- In plain English: $\Theta(g(n))$ are all functions $f(n)$ for which there exists three positive constants c_1, c_2 and n_0 such that for all $n \geq n_0$, $0 \leq c_1\, g(n) \leq f(n) \leq c_2\, g(n)$
  - $g(n)$ is an asymptotic **tight bound** for $f(n)$ (same growth rate)
- Intuitively, you can think of Ω as "=" for functions
- If $f(n) \in \Theta(g(n))$, we write $f(n) = \Theta(g(n))$

NOTE: The definitions imply a constant n_0 beyond which they are satisfied. We do not care about small values of n.

Asymptotic Notation – Theta, Θ

Theorem

$$f(n) = \Theta\big(g(n)\big) \text{ if } f(n) = O\big(g(n)\big) \text{ and } f(n) = \Omega\big(g(n)\big)$$

Asymptotic Notation – Theta, Θ

- Examples

$n^2 - 2n = \Theta(n^2)$?

$6n^3 = \Theta(n^2)$?

Asymptotic Notation – Theta, Θ

- Examples

$$n^2 - 2n = \Theta(n^2)$$

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

Asymptotic Notation – Theta, Θ

- Examples

Show that for any real constants a and b, where $b > 0$,

$$(n + a)^b = \Theta(n^b)$$

Names of Bounding Functions

$g(n) = O(f(n))$ means $C \times f(n)$ is an *upper bound* on $g(n)$.

$g(n) = \Omega(f(n))$ means $C \times f(n)$ is a *lower bound* on $g(n)$.

$g(n) = \Theta(f(n))$ means $C_1 \times f(n)$ is an upper bound on $g(n)$ and $C_2 \times f(n)$ is a lower bound on $g(n)$.

C, C_1, and C_2 are all constants independent of n.

Asymptotic Notation in Equations and Inequalities

- On the right-hand side alone of an equation (or inequality) $\equiv$ a set of functions

  Ex., $n = O(n^2) \leftrightarrow n \in O(n^2)$

- In general, in a formula, stands for some anonymous function that we do not care to name

  Ex., $2n^2 + 3n + 1 = 2n^2 + \Theta(n) \leftrightarrow 2n^2 + 3n + 1 = 2n^2 + f(n)$,

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

Next Class

Algorithm Analysis I (cont.)

Reading: Weiss, chap. 2