CS 170 DISCUSSION 1

ASYMPTOTIC ANALYSIS

Raymond Chan
UC Berkeley Fall 17

GETTING TO KNOW EVERYONE

- On index cards, write
  - Name
  - Where you're from
  - Something fun you did over summer
  - Something interesting (hobbies, interesting fact, …)
  - Anything else you would like to write down

ADMINISTRIVIA

- Course website: cs170.org
- My notes, links, and slides: raychan3.github.io/cs170/fa17.html
  - (Not live just yet)
- Sections:
  - 101: 9 - 10 am Etcheverry 3109
  - 108: 1 - 2 pm Hearst Field Annex B5
- Office Hours: Wed 1 - 2:30 pm Soda 411
- Email: raymondchan243@berkeley.edu

ASYMPTOTIC NOTATION

- Look at algorithm complexity when input is large.
- Notations: **O**, Ω, **Θ**
- Let *f(n)* and *g(n)* be function from positive integers to positive real numbers on inputs of size *n*.
- $f \in \mathbf{O}(g)$ if there is a constant $c > 0$ such that $f(n) \leq c \cdot g(n)$

ASYMPTOTIC NOTATION

- Look at algorithm complexity when input is large.
- Notations: **O**, Ω, **Θ**
- Asymptotic notations are for sets of functions.
- Algorithm runtimes expressed as a function of inputs of size *n*.
- If $f \in \mathbf{O}(n)$, then algorithm's runtime is in the set of functions $f \in \mathbf{O}(n)$
    - Also $f \in \mathbf{O}(n^2)$, $O(2^n)$ …

O (BIG-O)

- Notation: **O**

- Let $f(n)$ and $g(n)$ be function from positive integers to positive real numbers on inputs of size n.

- $f \in \mathbf{O}(g)$ if there is a constant $c > 0$ such that $f(n) \leq c \cdot g(n)$

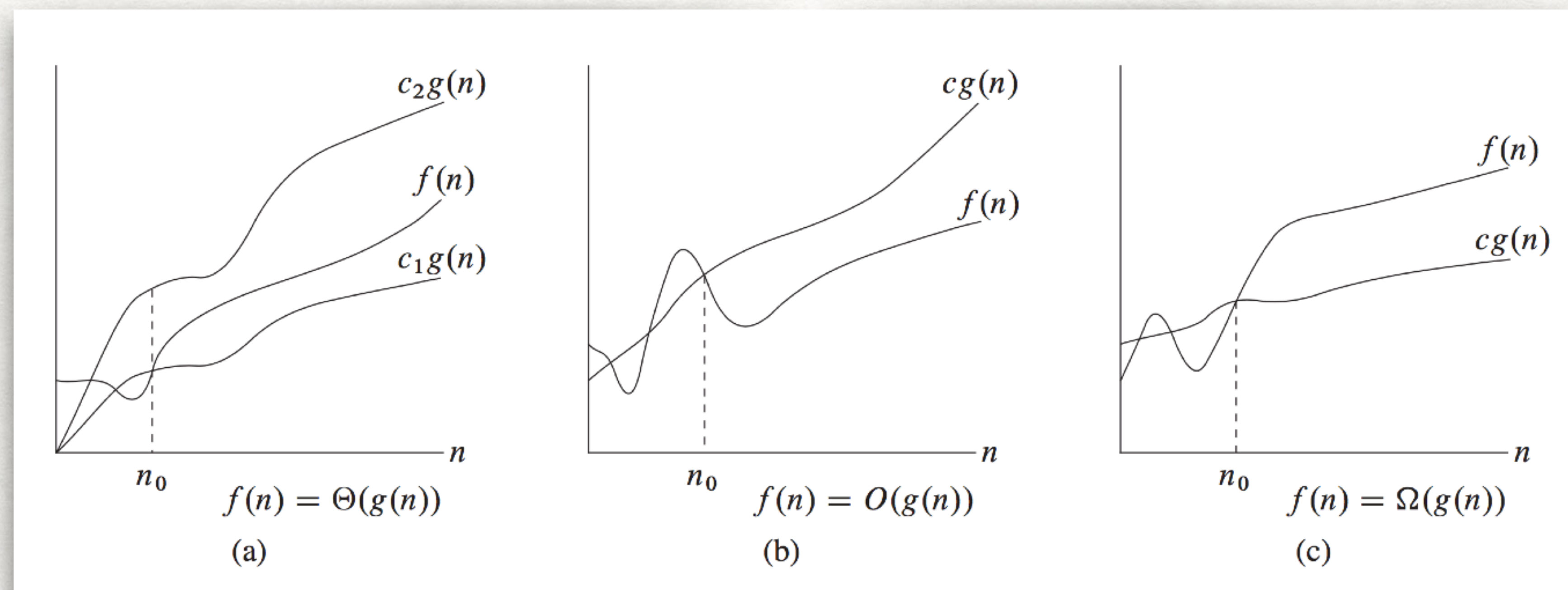
- $f(n)$ belongs to set of functions that are "upper-bounded" by $g(n)$ when <u>*n* gets significantly large</u>

    - $f \in O(n^2)$ and $g \in O(n^3)$, g dominates, but f could be slower.

    - $f = 1000n^2$ and $g = n^3$

O (BIG-O)

- *f(n)* belongs to set of functions that are "upper-bounded" by *g(n)* when <u>*n* gets significantly large</u>
  - $f \in O(n^2)$ and $g \in O(n^3)$, g dominates, but *f* could be slower.
  - ex. $f = 1000n^2$ and $g = n^3$



Introduction to Algorithms, Cormen, Leiserson, Rivest, Stein

Ω (BIG-OMEGA)

- Notation: Ω
- Let $f(n)$ and $g(n)$ be function from positive integers to positive real numbers on inputs of size n.
- $f \in \Omega(g)$ if there is a constant $c > 0$ such that $f(n) \geq c \cdot g(n)$
- $f(n)$ belongs to set of functions that are "lower-bounded" by $g(n)$ when <u>*n* gets significantly large</u>
- $g \in O(f)$
- If $f \in \Omega(n^3)$, then $f \in \Omega(n^2)$, $f \in \Omega(1)$ …

Θ (BIG-THETA)

- Notation: **Θ**
- Let *f(n)* and *g(n)* be function from positive integers to positive real numbers on inputs of size *n.*
- $f \in \Theta(g)$ if there is a constant $c_1 > 0$ and $c_2 > 0$ such that
  - $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ for all $c_1 \leq c_2$
- *f(n)* belongs to set of functions that are "tight-bounded" by *g(n)* when <u>*n* gets significantly large</u>
- $f \in \Omega(g)$ and $f \in O(g)$

COMMON ASYMPTOTIC SETS

- $O(1)$: constant
- $O(\log n)$: logarithmic
- $O(\sqrt{n})$: square root
- $O(n)$: linear
- $O(n \log n)$: `n logn`
- $O(n^2)$: quadratic
- $O(n^3)$: cubic
- $O(2^n)$: exponential
- $O(n!)$: factorial

thanks to allentang.me

TIPS AND TRICKS

- Remove multiplicative constants and lower order terms
  - $O(2n^4 + n^2 + n \log(n)) \in O(n^4)$
- Any exponential dominates any polynomial
  - $n^2 \in O(2^n)$
- Any polynomial dominates any logarithm
  - $n^2 \in \Omega(\log_3 n)$

TIPS AND TRICKS

- Removing exponents
  - $x^b = 2^{\log (x^\wedge b)} = 2^{b \log (x)}$

$$\lim_{n\to\infty} \frac{f}{g} = \begin{cases} 0 & f = O(g) \\ c, 0 < c < \infty & f = \Theta(g) \\ \infty & f = \Omega(g) \end{cases}$$

thanks to allentang.me

- Limits and ratio
  - *f(n) = n*
  - *g(n) = log n*
  - *f(n) ∈ Ω(g(n))*

$$\lim_{n\to\infty} \frac{f}{g} = \frac{n}{\log n} \to \frac{1}{n^{-1}} = n \to \infty$$

thanks to allentang.me

PROOFS

- 4 Part Algorithm Proofs by David Wagner
  - Main Idea
  - Pseudocode
  - Proof of Correctness
  - Runtime Analysis
- Stanford CS 161 Proof of Correctness