CSE 326: Data Structures
Lecture #3
June 23, 2000

Asymptotic Analysis (continued)

Terminology

Given an algorithm whose running time is $T(n)$

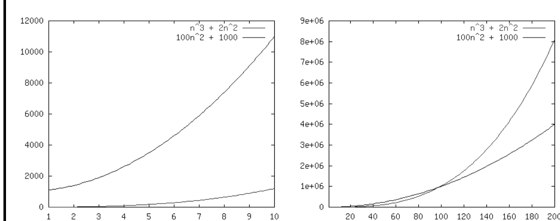
- $T(n) \in O(f(n))$ if there are constants c and n_0 such that $T(n) \le c\, f(n)$ for all $n \ge n_0$
  - $1, \log n, n, 100n \in O(n)$
- $T(n) \in \Omega(f(n))$ if there are constants c and n_0 such that $T(n) \ge c\, f(n)$ for all $n \ge n_0$
  - $n, n^2, 100 \cdot 2^n, n^3 \log n \in \Omega(n)$
- $T(n) \in \theta(f(n))$ if $T(n) \in O(f(n))$ and $T(n) \in \Omega(f(n))$
  - $n, 2n, 100n, 0.01\, n + \log n \in \theta(n)$
- $T(n) \in o(f(n))$ if $T(n) \in O(f(n))$ and $T(n) \notin \theta(f(n))$
  - $1, \log n, n^{0.99} \in o(n)$

Silicon Downs

Post #1	Post #2	Winner
$n^3 + 2n^2$	$100n^2 + 1000$	$O(n^2)$
$n^{0.1}$	$\log n$	$O(\log n)$
$n + 100n^{0.1}$	$2n + 10 \log n$	**TIE** $O(n)$
$5n^5$	$n!$	$O(n^5)$
$n^{-15}2^n/100$	$1000n^{15}$	$O(n^{15})$
mn^3	$2^m n$	**IT DEPENDS**

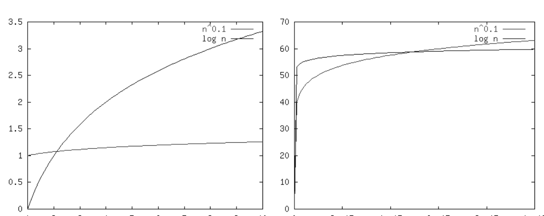
Race I

$n^3 + 2n^2$ vs. **$100n^2 + 1000$**



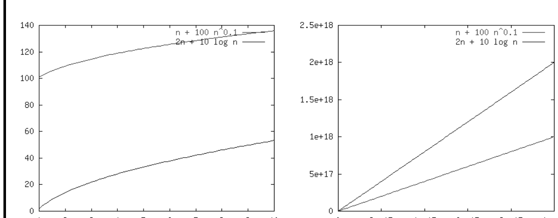
Race II

$n^{0.1}$ vs. **$\log n$**



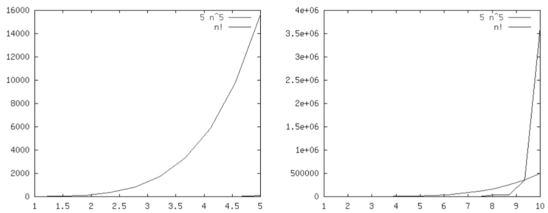
Race III

$n + 100n^{0.1}$ vs. **$2n + 10 \log n$**



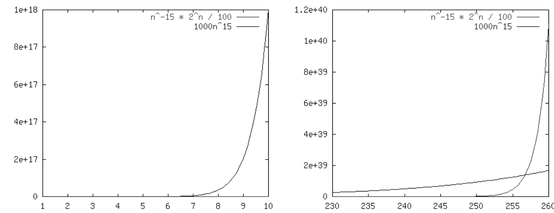
Race IV

$5n^5$ vs. $n!$



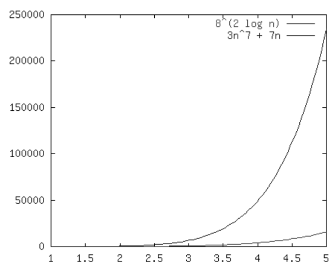
Race V

$n^{-15}2^n/100$ vs. $1000n^{15}$



Race VI

$8^{2\log(n)}$ vs. $3n^7 + 7n$



FBI Finds Silicon Downs Fixed

- The fix sheet (typical growth rates in order)
  - constant: $O(1)$
  - logarithmic: $O(\log n)$ ($\log_k n$, $\log n^2 \in O(\log n)$)
  - poly-log: $O(\log^k n)$
  - linear: $O(n)$
  - log-linear: $O(n \log n)$
  - superlinear: $O(n^{1+c})$ (c is a constant > 0)
  - quadratic: $O(n^2)$
  - cubic: $O(n^3)$
  - polynomial: $O(n^k)$ (k is a constant)
  - exponential: $O(c^n)$ (c is a constant > 1)

Types of analysis

Orthogonal axes
- bound flavor
  - upper bound (O, o)
  - lower bound (Ω, ω)
  - asymptotically tight (θ)
- analysis case
  - worst case (adversary)
  - average case
  - best case
  - "common" case
- analysis quality
  - loose bound (any true analysis)
  - tight bound (no better bound which is asymptotically different)

How Do We Justify Our Analysis?

- Code up programs and measure their behavior
  - Pro: concrete, observable
  - Con: may depend on individual computer or programmer skill or particular data set
- Techniques of mathematical proof
  - Pro: independent of individual computer, programmer skill or particular data set
  - Con: not always easy

Common Proof Techniques

- Counterexample
  - show an example which does not fit with the theorem
  - QED (the theorem is disproved)
- Contradiction
  - assume the opposite of the theorem
  - derive a contradiction
  - QED (the theorem is proven)
- Induction

  Step 1. prove for a base case (e.g., n = 1)

  Step 2. assume true for all values through some anonymous value (n)

  Step 3. prove for the next value (n + 1)

  Step 4. QED

  Dickey's Step –1: Convince yourself it's true!

Example for Induction Proof

- What is the sum of the 1st N integers?

Another Induction Example

- A number is divisible by 3 iff the sum of its digits is divisible by three
- Step –1: What is the theorem saying? Is it really true?
- Base case(s):
- General case(s):

Reading for Next Reading Quiz

- **Review Chapter 2**
- **Chapter 3.1-3.2**