**University of Virginia** 5 October 2011
**cs1120: Introduction of Computing**
**Explorations in Language, Logic, and Machines**

# Class 19: Analyzing Algorithms

## Upcoming Schedule

- **Assistant Coaches' Review Sessions for Exam 1:**
  - Tuesday, 6:30pm, Rice 442
  - Wednesday (today), 7:30pm, Rice 442
- **Office Hours this week:**
  - **Dave:** (in Rice 507) Monday, 1:15-2pm; Tuesday, 11am-noon; Thursday, 9:45am-11
  - No assistant coaches' office hours on Monday, Tuesday, or Wednesday
  - Thursday: (in Rice Bagel Space): 1-2:30pm (Joseph), 4:30-6pm (Jonathan), 6-7:30pm (Jiamin)
  - No scheduled office hours while exam is out (Friday 7 October – Wednesday 12 October)
- **Wednesday, 12 October:** Exam 1 Due (will be take-home and open book). You will be permitted to use any non-human resources you want for Exam 1, other than using DrRacket (or any other Scheme interpreter). Exam 1 will be handed out on **Friday, 7 October**. It covers:
  - Problem Sets 1-4 including PS Comments 1-4
  - Course Book Chapters 1-6
  - Classes 1-18 (but not the new material on cost and running time)

## Asymptotic Operators

***O* (Big-Oh)**: *upper bound*. A function $g$ is in *O (f)* iff there are positive constants $c$ and $n_0$ such that $g(n) \leq cf(n)$ for all $n \geq n_0$.

**Ω (Omega)**: *lower bound*. A function $g$ is in *Ω (f)* iff there are positive constants $c$ and $n_0$ such that $g(n) \geq cf(n)$ for all $n \geq n_0$.

**Θ (Theta)**: *tight bound*. A function $g$ is in $\Theta(f)$ iff $g$ is in *O(f)* and $g$ is in *Ω(f)*.

Prove $1/1000\ n^2 - 9999n^{1.9} \in \Theta(n^2)$

Prove $\Theta(1/1000\ n^2 - 9999n^{1.9}) = \Theta(n^2)$

What is the asymptotic running time of:

```
(define (bigger a b)
  (if (> a b) a b))
```

Implementing > (greater than) using a Turing Machine, unary notation for inputs:

$\mathbf{1}^a\mathbf{>1}^b\mathbf{\#} \Rightarrow \mathbf{0\#}$ if $a > b$, $\mathbf{1\#}$ if $a \leq b$

Turing Machine transition rules: (*state*, *read symbol*) $\rightarrow$ (*next state*, *write symbol*, *direction to move*)

Implementing > (greater than) using a Turing Machine, *binary* notation for inputs: