University of Virginia 3 October 2011
cs1120: Introduction of Computing
Explorations in Language, Logic, and Machines

Class 18: Measuring Cost

Upcoming Schedule

- **Now:** Problem Set 4
- **Assistant Coaches' Review Sessions for Exam 1:**
  - Tuesday (tomorrow), 6:30pm, Rice 442
  - Wednesday, 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). Exam 1 will be handed out on **Friday, 7 October**. Exam 1 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)

Analyzing Cost

Why do we use Turing Machines to measure cost abstractly, rather than using a "real" computer?

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 $\Omega(f)$ 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 $\Omega(f)$.

Why are these operators useful for understanding the cost of evaluating procedures?

Think of $O(n)$ as O(lambda (n) n) and $O(n^2)$ as O(lambda (n) (* n n)).

Draw a diagram showing: n^3, $O(n)$, $O(n^2)$, $\Omega(n^2)$, $\Theta(n^2)$, and $O(1)$.

Prove n is in $O(n^2)$.

Prove n is not in $\Omega(n^2)$.

Prove n is not in $\Theta(n^2)$.

Prove $7n + 13$ is in $\Theta(n)$.