# CS3000: Algorithms & Data
# Jonathan Ullman

Lecture 1:
- Course Overview
- Warmup Exercise (Induction, Asymptotics, Fun)

Sep 7, 2018

# Me

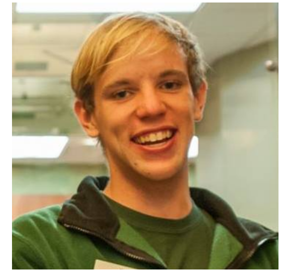
- Name: Jonathan Ullman
  - Call me Jon
  - NEU since 2015
  - Office: 623 ISEC
  - Office Hours: Wed 10:30-12:00

- Research:
  - Privacy, Cryptography, Machine Learning, Game Theory
  - Algorithms are at the core of all of these!

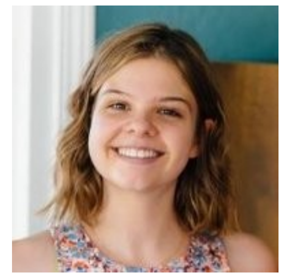
# The TA Team

OH 2:30-6:30 on Mondays
OH 12:00-4:00 on Thursdays
OH ??? on Wednesdays

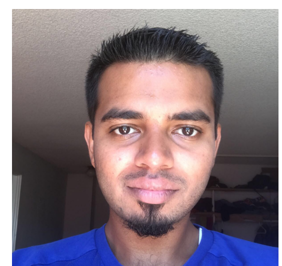
- **Jerry Lanning**
  - Office Hours: TBD
  - Location: TBD



- **Lisa Oakley**
  - Office Hours: Thu 12:00-2:00
  - Location: TBD

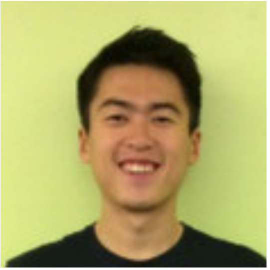


- **Chandan Shankarappa**
  - Office Hours: Mon 2:30-4:30
  - Location: TBD



# The TA Team

- **Tian Xia**
  - Office Hours: Thu 2:00-4:00
  - Location: TBD



- **Lydia Zakynthinou**
  - Office Hours: Mon 4:30-6:30
  - Location: TBD

# Algorithms

- What is an algorithm?

  *An explicit, precise, unambiguous, mechanically-executable sequence of elementary instructions for solving a computational problem.*

  *-Jeff Erickson*

  - Essentially all computer programs (and more) are algorithms for some computational problem.

# Algorithms

- sort this list of numbers
- find the shortest route home
- find websites about algorithms

- What is Algorithms?

*The study of how to solve computational problems.*

- Abstract and formalize computational problems
- Identify broadly useful algorithm design principles for solving computational problems
- Rigorously analyze properties of algorithms
  - This Class: correctness, running time, space usage
  - Beyond: extensibility, robustness, simplicity,…

# Algorithms

- What is CS3000: Algorithms & Data?

  *The study of how to solve computational problems.*
  *How to rigorously prove properies of algorithms.*

  - Proofs are about understanding and communication, not about formality or certainty
    - Different emphasis from courses on logic
    - We'll talk a lot about proof techniques and what makes a correct and convincing proof

# Algorithms

- That sounds hard.  Why would I want to do that?

- Build Intuition:
  - How/why do algorithms really work?
  - How to attack new problems?
  - Which design techniques work well?
  - How to compare different solutions?
  - How to know if a solution is the best possible?

# Algorithms

- That sound hard. Why would I want to do that?

- Improve Communication:
  - How to explain solutions?
  - How to convince someone that a solution is correct?
  - How to convince someone that a solution is best?

# Algorithms

- That sounds hard.  Why would I want to do that?

- Learn Problem Solving / Ingenuity
  - “Algorithms are little pieces of brilliance…” -Olin Shivers

# Algorithms

- That sounds hard.  Why would I want to do that?

- Get Rich:
  - Many of the world's most successful companies (notably Google) began with algorithms.
- Understand the natural world:
  - Brains, cells, networks, etc. often viewed as algorithms.
- Fun:
  - Yes, seriously, fun.

# Algorithms

- That sounds hard. Why would I want to do that?
- You can only gain these skills with practice!

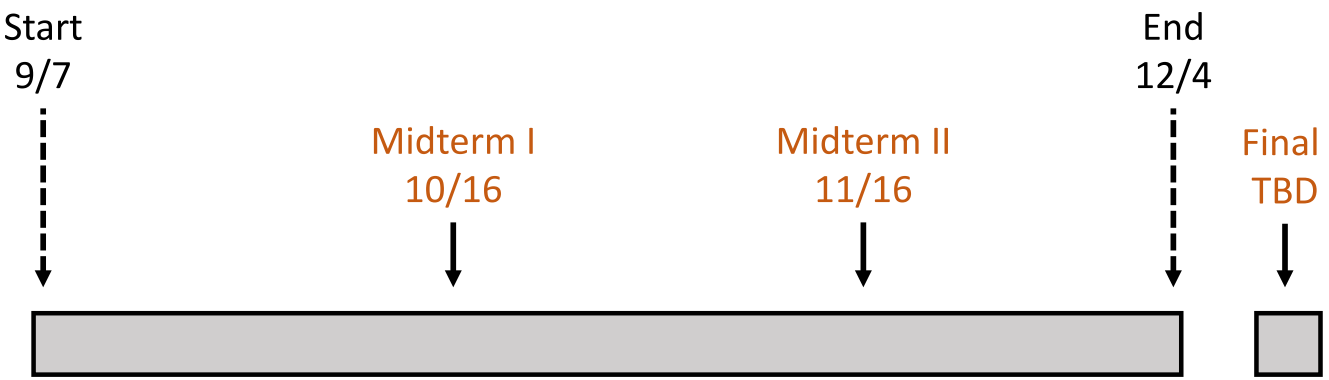
OH

HW

Textbook

Other resources online

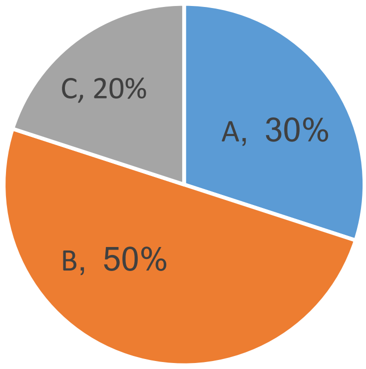
# Course Structure



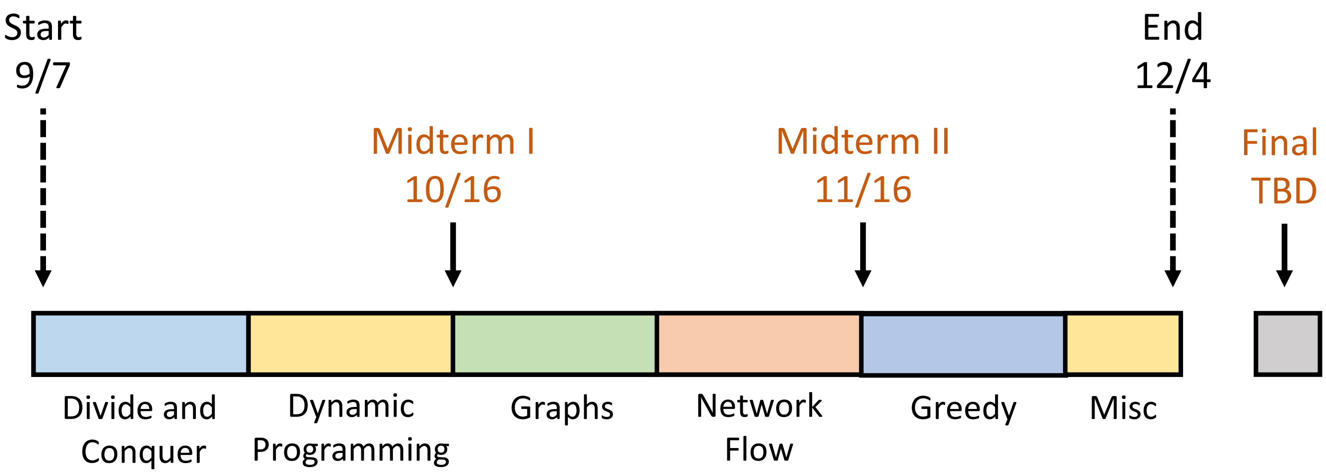

7-10

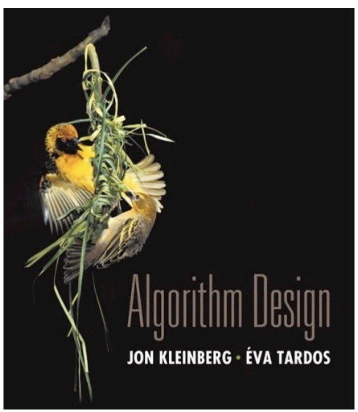
- HW = 45%
- Exams = 55%
  - Midterm I = 15%
  - Midterm II = 15%
  - Final = 25%

Typical Grade Distribution



# Course Structure




Textbook:
Algorithm Design by Kleinberg and Tardos

More resources on the course website

# Homework

- Weekly HW Assignments (45% of grade)
  - Due Tuesdays by 11:59pm
  - **HW1 out now!  Due Tue 9/18**
  - No extensions, no late work
  - Lowest HW score will be dropped from your grade

- A mix of mathematical and algorithmic questions

# Homework Policies

- Homework must be typeset in LaTeX!
  - Many resources available
  - Many good editors available (TexShop, TexStudio)
  - I will provide HW source

on Mac

**The Not So Short**
**Introduction to LaTeX $2_\varepsilon$**

*Or LaTeX $2_\varepsilon$ in 157 minutes*

**by Tobias Oetiker**
**Hubert Partl, Irene Hyna and Elisabeth Schlegl**

Version 5.06, June 20, 2016

# Homework Policies

- Homework will be submitted on Gradescope!
  - Entry code: 94V4YJ
  - Sign up today, or even right this minute!

gradescope

- use gradescope for regrades

# Homework Policies

- You are encouraged to work with your classmates on the homework problems.
  - You may not use the internet
  - You may not use students/people outside of the class

- **Collaboration Policy:**
  - You must write all solutions by yourself
  - You may not share any written solutions
  - You must state all of your collaborators
  - We reserve the right to ask you to explain any solution

# Course Website

http://www.ccs.neu.edu/home/jullman/cs3000f18/syllabus.html
http://www.ccs.neu.edu/home/jullman/cs3000f18/schedule.html

## CS3000: Algorithms & Data

Syllabus Schedule

**This schedule will be updated frequently—check back often!**

| # | Date | Topic | Reading | HW |
|---|---|---|---|---|
| 1 | F 9/7 | Course Overview | --- | HW1 Out (pdf, tex) |
| 2 | T 9/11 | Stable Matching: Gale-Shapley Algorithm | KT 1.1,1.2,2.3 | --- |
| 3 | F 9/14 | Divide and Conquer: Mergesort, Asymptotic Analysis | KT 5.1, 2.1-2.2 | |
| 4 | T 9/18 | Divide and Conquer: Karatsuba, Recurrences | KT 5.2, 5.5<br>Erickson II.1-3 | **HW1 Due**<br>HW2 Out |
| 5 | F 9/21 | Divide and Conquer: Master Theorem, Median | Erickson 1.5-1.7 | |
| 6 | T 9/25 | Divide and Conquer: More Examples | --- | **HW2 Due** |

Note to self: Fix Piazza

# Discussion Forum

- We will use Piazza for discussions
  - Ask questions and help your classmates
  - Please use private messages sparingly
- Sign up today, or even right this minute!

# What About the Other Sections?

- I teach two sections: TF 1:35 and TF 3:25
  - These sections are completely interchangeable
  - You may collaborate on HW across my sections
  - You may go to OH for any of my TAs

- Prof. Neal Young teaches another section
  - No formal relationship with my sections
  - Will cover very similar topics and share some materials
  - You may not collaborate with Prof. Young's section
  - You should not go to OH for Prof. Young's TAs

One More Thing:
I need to count how many students are in this class!

# Simple Counting

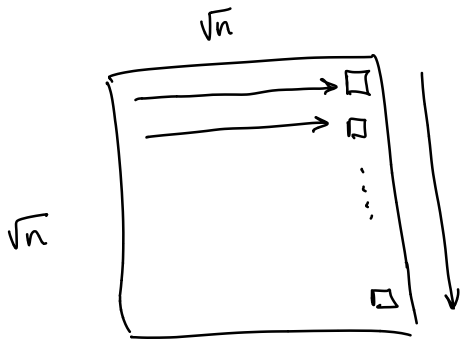
74 students

43.16 seconds

```
SimCount:
  Find first student
  First student says 1
  Until we're out of students:
    Go to next student
    Next student says (what last student said + 1)
```

- Is this correct? Yes.
- How long does this take with $n$ students?

$$T(n) = 2n$$




$$T(n) = 2\sqrt{n} + 2\sqrt{n}$$

$$= 4\sqrt{n}$$

$4\sqrt{n}$ beats $2n$ if $n > 4$

# Recursive Counting

70 student

1:10.73

```
RecCount:
  Everyone set your number to 1
  Everyone stand up
  Until only one student is standing:
    Pair up with a neighbor, wait if you don't find one
    For each pair:
      Sum up your numbers
      Sit down if you are the taller person in the pair
  Say your number
```

- Is this correct? Why?

Loop invariant

After afte every iteration of the loop, the sum of the # of everyone standing is n.

# Recursive Counting

```
RecCount:
  Everyone set your number to 1
  Everyone stand up
  Until only one student is standing:
    Pair up with a neighbor, wait if you don't find one
    For each pair:
      Sum up your numbers
      Sit down if you are the taller person in the pair
  Say your number
```

- How long does this take with $n$ students?

$T(n)$ = time RecCount takes with $n$ students

$T(n) = 2 + T(\lceil \frac{n}{2} \rceil)$ $\quad T(1) = 3$

# Running Time

- **Recurrence:** $T(1) = 3, T(n) = 2 + T(\lceil n/2 \rceil)$

$$T(1) = 3$$

$$T(2) = 2 + T(1) = 2 + 3 = 5$$

$$T(4) = 2 + T(2) = 2 + 2 + T(1) = 7$$

$$T(2^m) = \underbrace{2 + 2 + \cdots + 2}_{m} + T(1) = 2m + 3$$

$$T(n) = 2\log_2 n + 3$$

# Running Time

- **Claim:** For every number of students $n = 2^m$
  $T(2^m) = 2m + 3$

# Proof by Induction

$T(n) = 2 + T(\frac{n}{2})$

$T(1) = 3$

- **Claim:** For every number of students $n = 2^m$ $T(2^m) = 2m + 3$

$\forall m \in \mathbb{N} \; T(2^m) = 2m + 3$

- **Induction:** "automatically" prove for every $m$
  - **Inductive Hypothesis:** Let $H(m)$ be the statement $T(2^m) = 2m + 3$

Claim $\iff \forall m \in \mathbb{N} \; H(m)$ is true

  - **Base Case:** $H(1)$ is true    $T(2) = 5$
  - **Inductive Step:** For every $m \geq 1$, $H(m) \Longrightarrow H(m+1)$
  - **Conclusion:** statement is true for every $m$

$\forall m \in \mathbb{N}$ if $T(2^m) = 2m + 3$ then $T(2^{m+1}) = 2(m+1) + 3$

# Proof by Induction

- **Claim:** For every number of students $n = 2^m$ $T(2^m) = 2m + 3$

IH: $H(m)$ : $T(2^m) = 2m + 3$

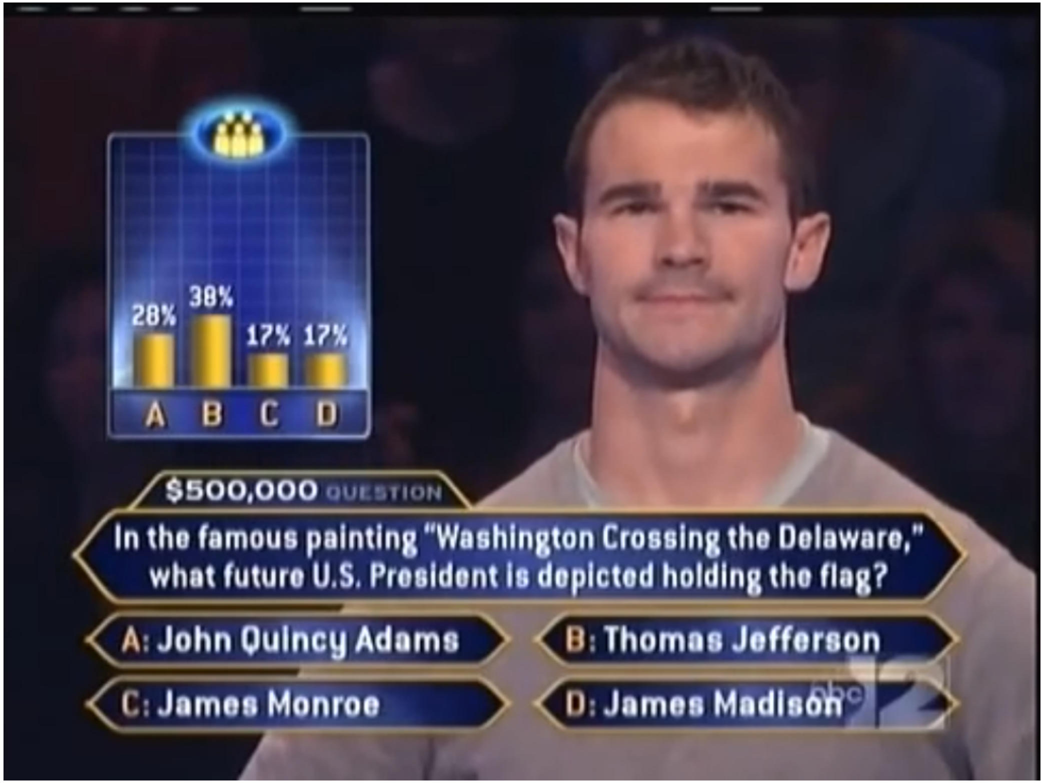
BC: $T(2) = 2 + T(1) = 5$ $\therefore$ $H(1)$ is true

IS: Pick any $m$, assume $H(m)$

$$T(2^{m+1}) = 2 + T(2^m)$$
$$= 2 + (2m + 3)$$
$$= 2(m+1) + 3$$

Uses the fact that $H(m)$ is true

# Ask the Audience




*Who Wants to be a Millionaire?*

# Ask the Audience

- **Claim:** For every $n \in \mathbb{N}$, $\sum_{i=0}^{n-1} 2^i = 2^n - 1$

- **Proof by Induction:**

# Running Time

$n = 74$

$2 \times 74 = 148$ steps

$2 \times \log_2(74) + 3 \leq 17$ steps

- **Simple counting:** $T(n) = 2n$ steps ≈ 43 seconds
- **Recursive counting:** $T(n) = 2\log_2 n + 3$ steps ≈ 70 seconds

- But for this class, simple counting was faster???

"steps" are not well defined

# Running Time

- **Simple counting:** $T(n) = 2n$ seconds
- **Recursive counting:** $T(n) = 30\log_2 n + 45$ seconds

- Compare algorithms by asymptotics!
  - Log-time beats linear-time as $n \to \infty$