CS 3000: Algorithms & Data
Jonathan Ullman

Lecture 1

- Course Overview
- Warmup Exercise (Induction, Asymptotics, Fun)

Jan 6, 2020

Me

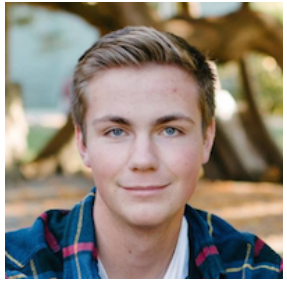
- **Jon Ullman**
  - Call me Jon
  - NEU since 2015
  - Office: 623 ISEC
  - Office Hours: Tue 2:30 – 4:00pm

- Research:
  - Privacy, Machine Learning, Cryptography, Game Theory
  - The common thread is **algorithms**

The TA Team

- **Danish Farooq**
  - Office Hours: Thu 6:00 – 8:00pm
  - Location: 605 ISEC

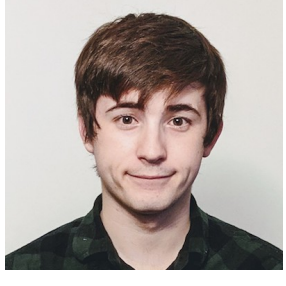
- **Zach Galeaz**
  - Office Hours: Fri 4:00 – 6:00pm
  - Location: 501 ISEC



The TA Team

- **Matthew Jones**
  - Office Hours: Thu 4:00 – 6:00pm
  - Location: 605 ISEC

- **Josh Towner**
  - Office Hours: Wed 5:00 – 7:00pm
  - Location: 605 ISEC



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

- What is algorithms?

  The study of how to solve computational problems.

  - Abstract and formalize computational problems
  - Identify useful algorithmic tools for solving computational problems
  - Analyze and compare algorithms
    - This Class: correctness, running time, space usage
    - Beyond: parallelism, data, robustness, simplicity, extensibility

Algorithms

- What is CS 3000: Algorithms & Data?

  The study of how to solve computational problems, and rigorously prove properties of algorithms

  - Proofs are about comprehension 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

Why would we do this?

- **Improve problem solving:**
  - 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?

Why would we do this?

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

Why would we do this?

- **Get Rich:**
  - Many of the world's most successful companies (e.g. Google, Akamai) began with algorithms.
  - These companies want you to solve algorithms problems on the spot

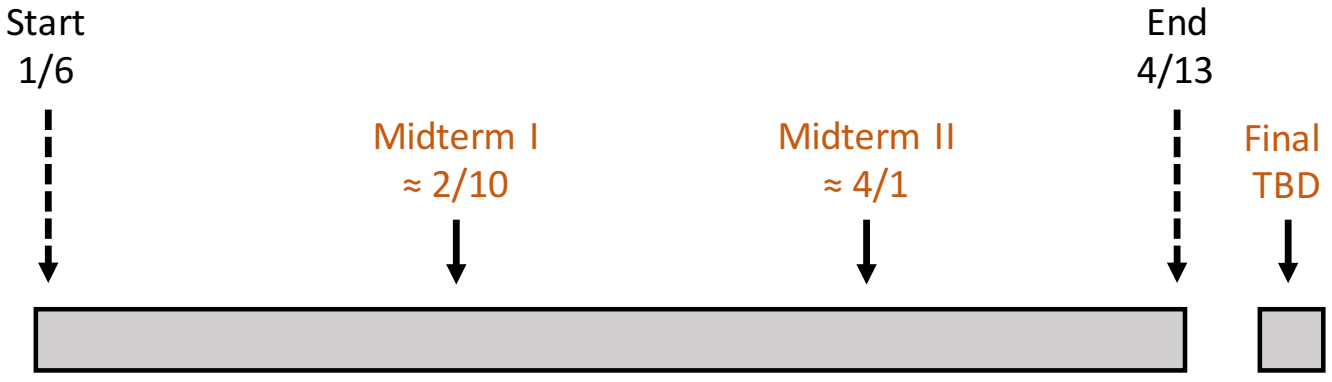
Why would we do this?

- **Understand the natural world:**
  - Brains, cells, networks, etc. often viewed as algorithms.
- **Exercise the brain:**
  - *"Algorithms are little packets of brilliance."* -Olin Shivers
- **Fun:**
  - Yes, seriously, fun.

Why would we do this?

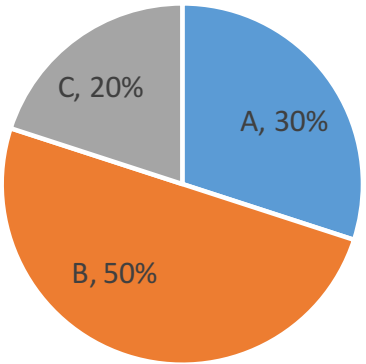
- **Learning these skills is hard, and takes practice!**

Course Structure

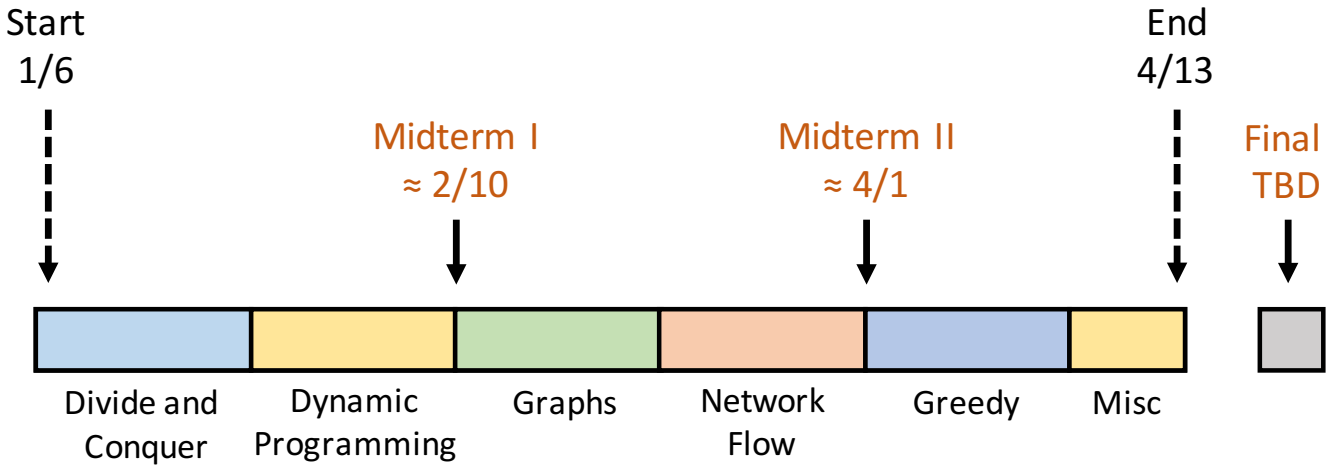


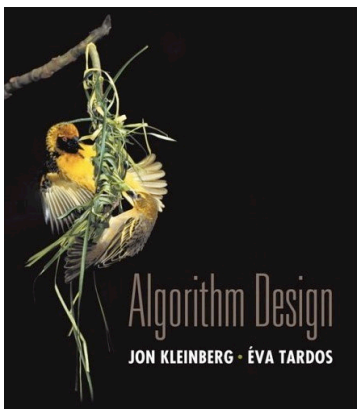
- HW = 45%
- Exams = 55%
  - Midterm I = 15%
  - Midterm II = 15%
  - Final = 25%

Typical Grade Distribution



Course Structure





Textbook:
Algorithm Design by Kleinberg and Tardos

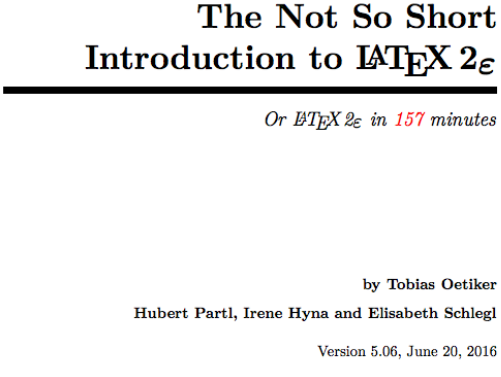
More resources on the course website

Homework

- Weekly HW Assignments (45% of grade)
  - Due Fridays by 11:59pm
  - **HW1 out now! Due Fri 1/17!**
  - 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



Homework Policies

- Homework will be submitted on Gradescope!
  - Entry code: **9GZP3P**
  - Sign up today, or even right this minute!

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

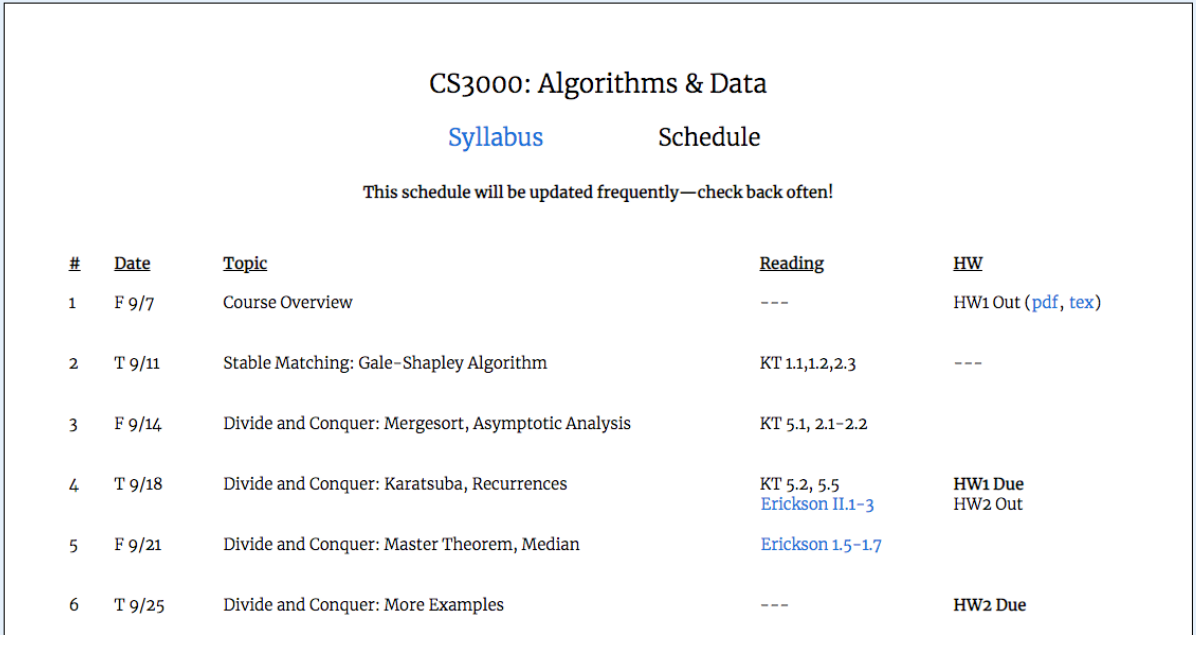
- **Collaboration Policy:**
  - You must write all solutions by yourself
  - You may not share any written solutions
  - You must state all of your collaborators
  - We reserve the right to ask you to explain any solution

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!

Course Website

http://www.ccs.neu.edu/home/jullman/cs3000s20/syllabus.html
http://www.ccs.neu.edu/home/jullman/cs3000s20/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 Erickson II.1-3	**HW1 Due** 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**

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

Simple Counting

73 students
57 seconds

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

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

Number of steps $\approx n$ (2n steps)

Fancy Counting

75 students?

178 seconds

```
FancyCount:
  Everyone set your number to 1
  Everyone stand up
  Until only one student is standing:
    Try to pair up with a neighbor
    If (you are not in a pair):
      Stay standing
    Else If (you are in a 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: Sum of the numbers of standing people is n.

Parallel Counting

```
FancyCount:
  Everyone set your number to 1
  Everyone stand up
  Until only one student is standing:
    Try to pair up with a neighbor
    If (you are not in a pair):
      Stay standing
    Else If (you are in a 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?

Running Time

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

Claim: $T(n) \approx \log_2(n)$

Running Time

- **Claim:** For every number of students $n = 2^m$
  $T(2^m) = 2m + 3$ $\quad T(n) = 2 \cdot \log_2(n) + 3$

$\forall\, m \in \mathbb{N} \quad T(2^m) = 2m + 3 \quad (\forall\, m \in \mathbb{N} \quad H(m) \text{ is true})$

Break it up into infinitely many

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

- $H(1)$ is true
- $H(m) \Rightarrow H(m+1) \quad \forall\, m \in \mathbb{N}$

Proof by Induction

- **Claim:** For every number of students $n = 2^m$ $T(2^m) = 2m + 3$ $\forall m \in \mathbb{N} \quad T(2^m) = 2m+3$

- **Induction:** "automatically" prove for every m
  - **Inductive Hypothesis:** Let $H(m)$ be the statement $T(2^m) = 2m + 3$
  - **Base Case:** $H(1)$ is true
  - **Inductive Step:** For every $m \geq 1$, $H(m) \Longrightarrow H(m+1)$
  - **Conclusion:** statement is true for every m

Inductive Step: $H(m) \Rightarrow H(m+1)$

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

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

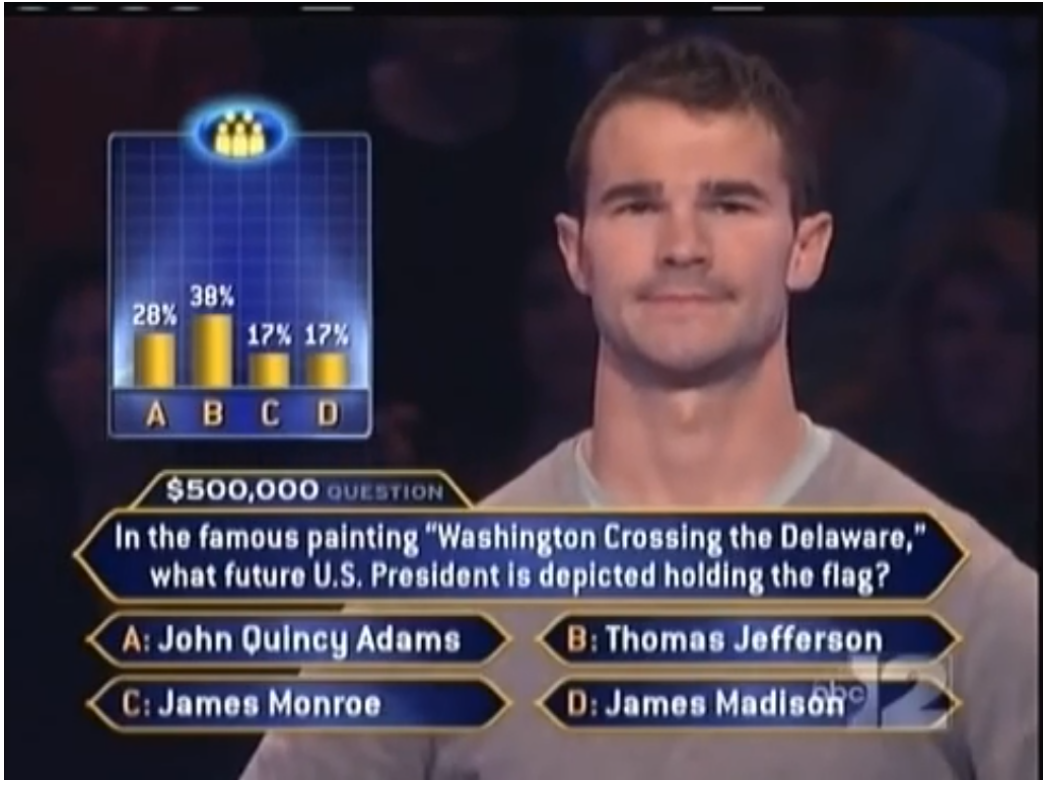
② $= 2 + 2m + 3$ (By IH)

③ $= 2(m+1) + 3$ (Algebra)

Proof by Induction

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

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

- **Simple counting:** $T(n) = 2n$ steps 57s
- **Fancy counting:** $T(n) = 2\log_2 n + 3$ steps 178s

- But for this class, simple counting was faster?

Running Time

$n = 70$

- **Simple counting:** $T(n) = 2n$ seconds $\approx 140s$
- **Fancy counting:** $T(n) = 30 \log_2 n + 45$ seconds $\approx 225s$

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