CS3000: Algorithms & Data
Paul Hand

Day 1:

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

Jan 7, 2019

Instructor

- Name: Paul Hand
  - Call me Paul
  - NEU since Fall 2018
  - Office: 523 Lake
  - Office Hours: Mon 1:15-2:45

- Research:
  - Machine Learning, Artificial Intelligence, Computer Vision
  - Algorithms are at the core of all of these!

Discussion:

What impressive things can computers now do (but couldn't when you were born)?

What do you think computers will be able to do in 10 years that they can't today?

Discussion:

What would you say is driving computational advances?

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 broadly useful algorithm design principles for solving computational problems
  - Rigorously analyze properties of algorithms
    - correctness, running time, space usage

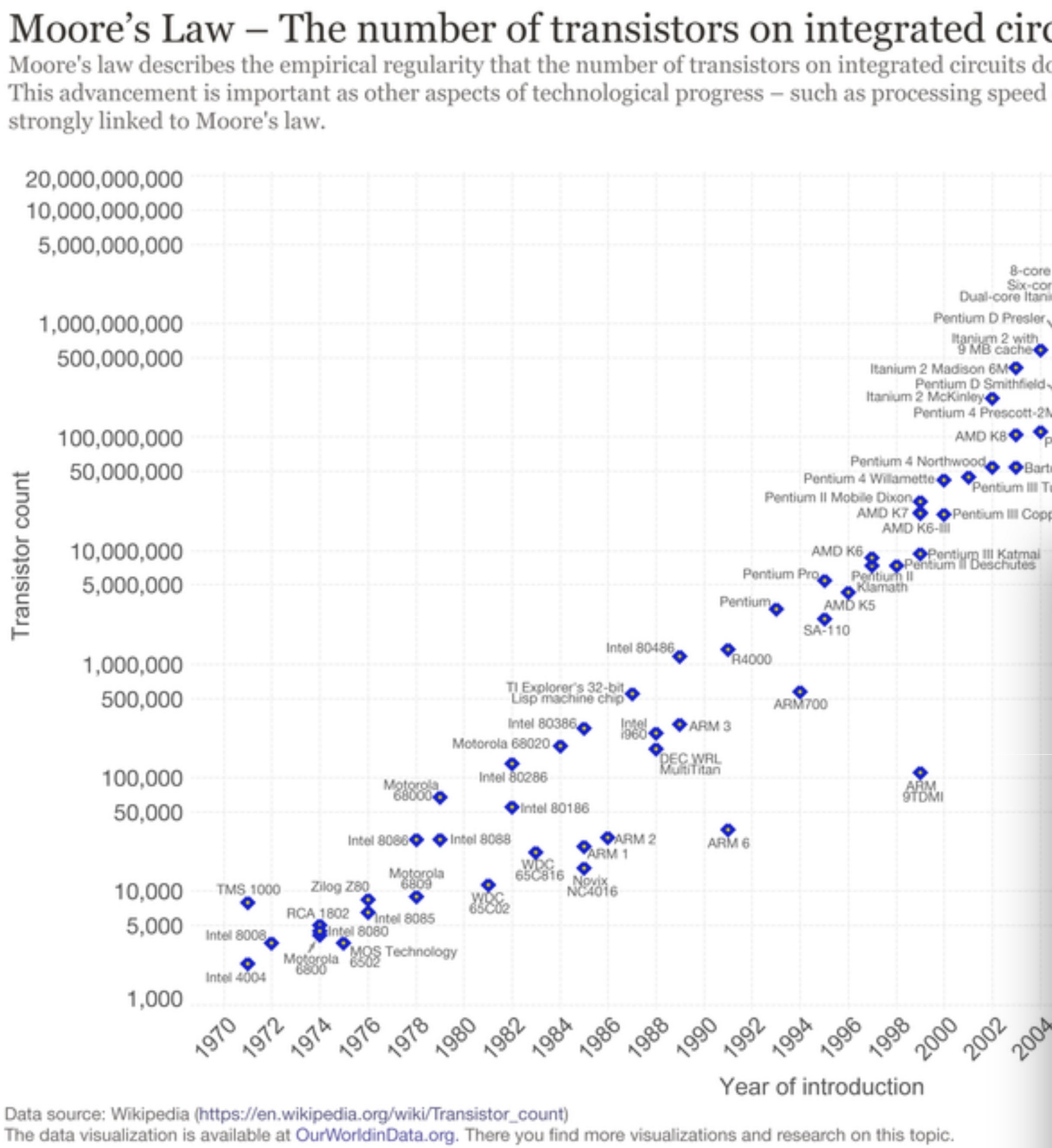
ARCHIVES | 1997

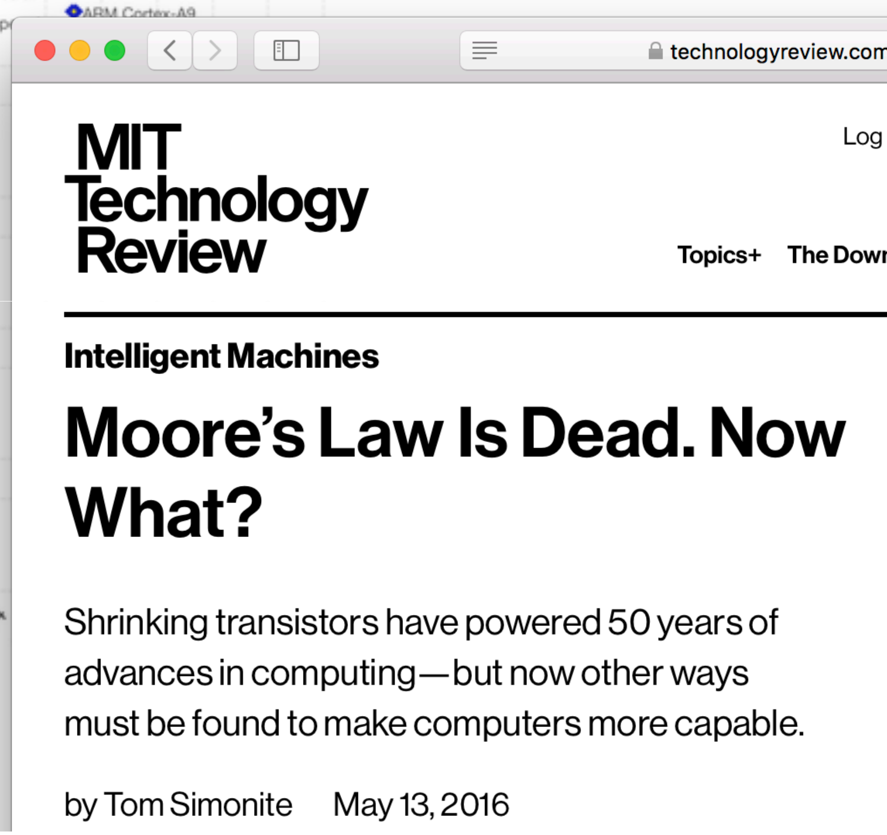
Swift and Slashing, Computer Topples Kasparov

By BRUCE WEBER MAY 12, 1997

In brisk and brutal fashion, the I.B.M. computer Deep Blue unseated humanity, at least temporarily, as the finest chess playing entity on the planet yesterday, when Garry Kasparov, the world chess champion, resigned the sixth and final game of the match after just 19 moves, saying, "I lost my

Moore’s Law




technologyreview.com

MIT Technology Review

Log

Topics+ The Down

Intelligent Machines

Moore’s Law Is Dead. Now What?

Shrinking transistors have powered 50 years of advances in computing—but now other ways must be found to make computers more capable.

by Tom Simonite May 13, 2016

Support The Guardian

Contribute → Subscribe →

Search jobs | Sign in | Search | US edition

The Guardian

News | Opinion | Sport | Culture | Lifestyle | More

US World Environment Soccer US Politics Business **Tech** Science

Artificial intelligence (AI)

AlphaGo seals 4-1 victory over Go grandmaster Lee Sedol

DeepMind's artificial intelligence astonishes fans to defeat human opponent and offers evidence computer software has mastered a major challenge

Steven Borowiec

Tue 15 Mar 2016 06.12 EDT

A computational problem

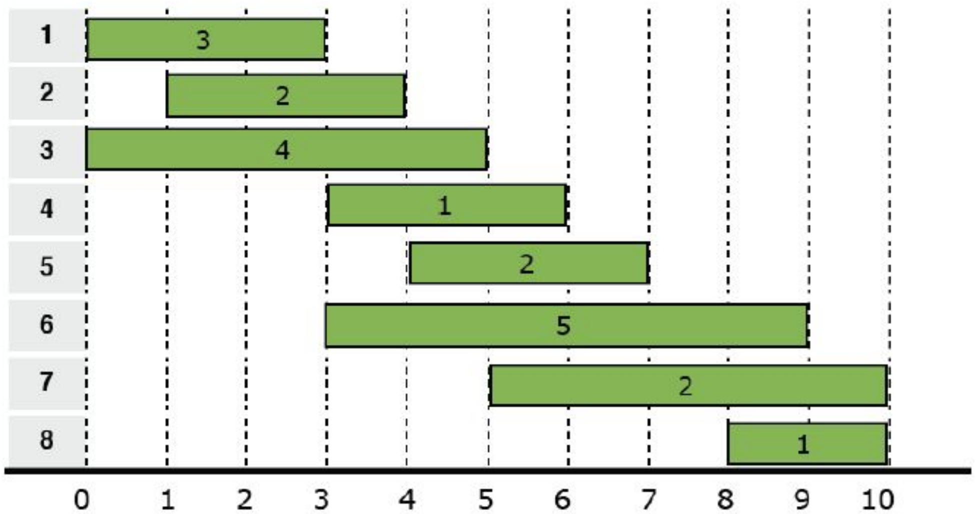
- Each of you:

  Determine how many other people in this room have the same first name as you.

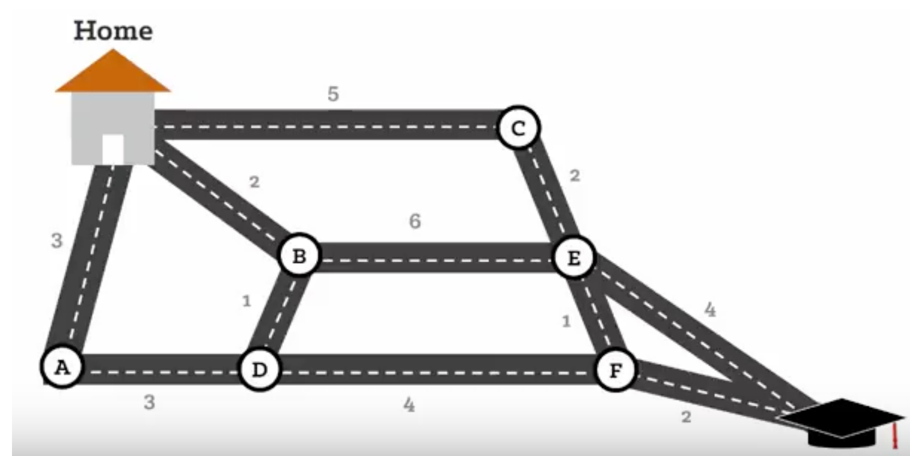
There are multiple ways to solve problems

- Sometimes your first instinct approach is reasonable.
- Sometimes your first instinct is not.

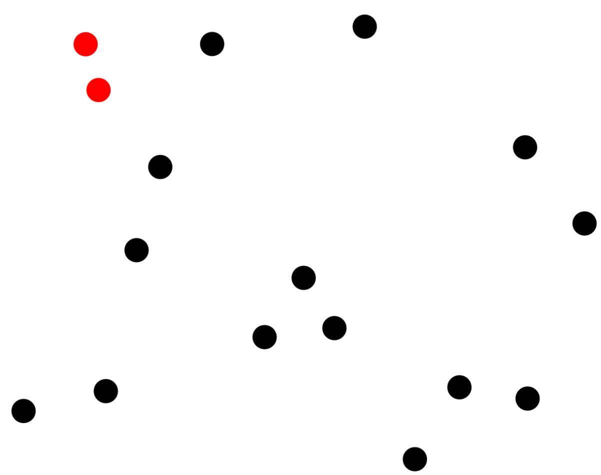
Example problems we will look at



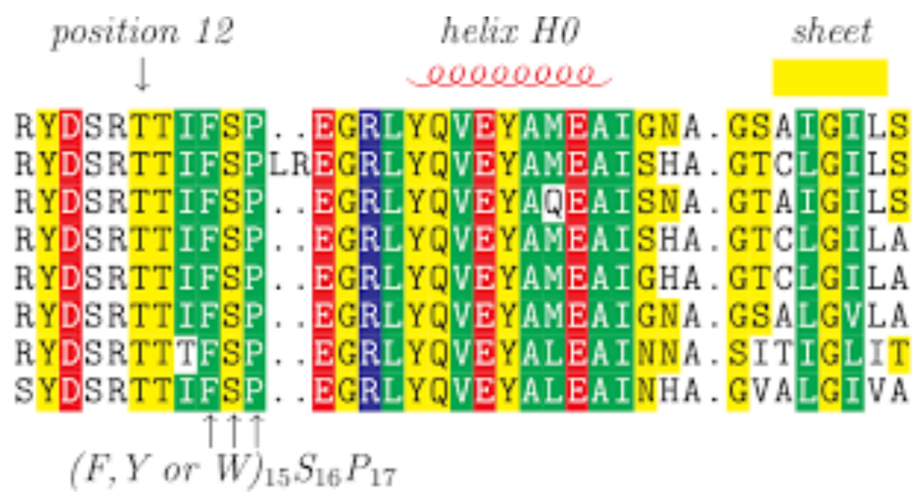
Interval Scheduling



Finding the shortest path

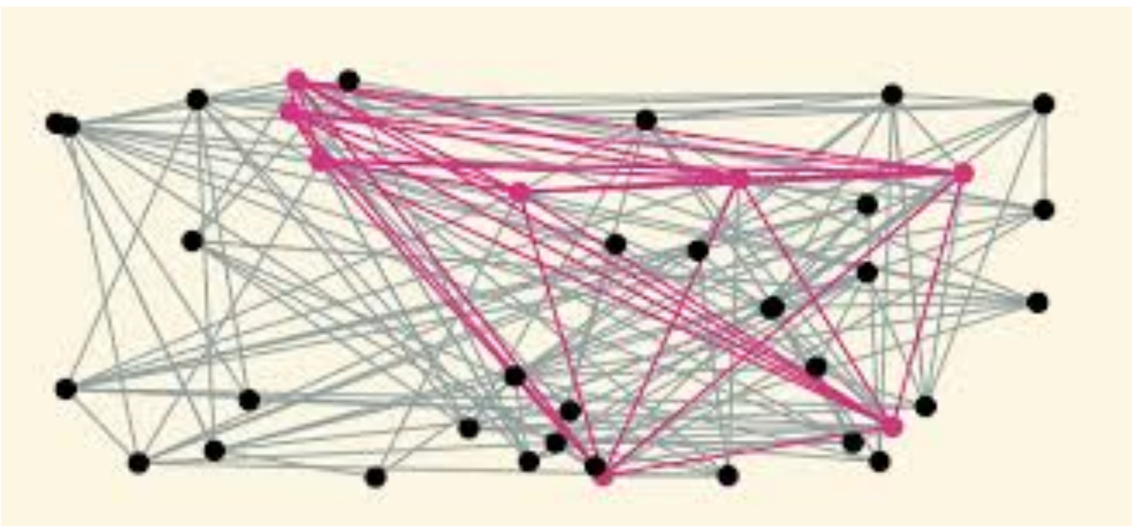


Closest Pair of Points

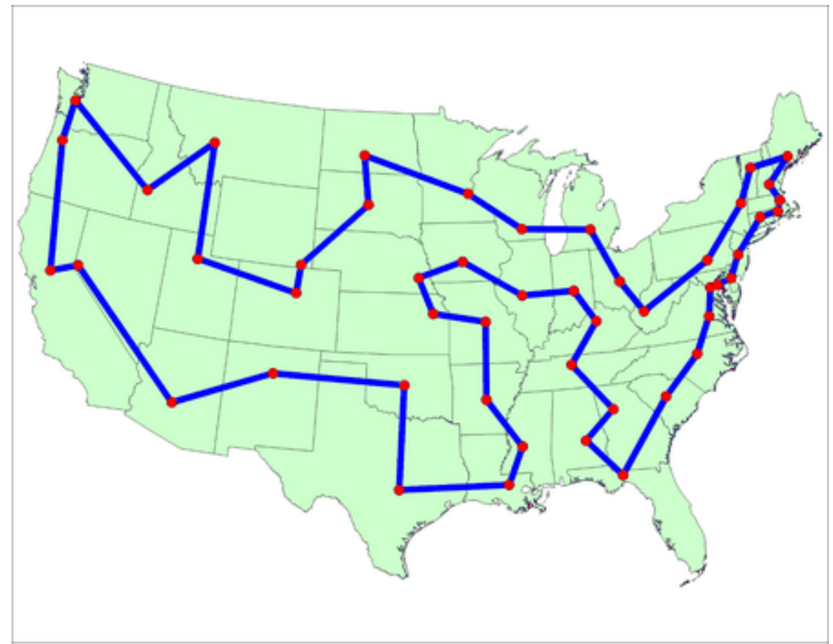


Sequence Alignment

Other good problems (we wont see)



Planted Clique



Traveling Salesman Problem

Algorithms

- What is CS3000: Algorithms & Data?

  The study of how to solve computational problems.
  How to rigorously prove properties 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 sounds hard. Why would I want to do that?

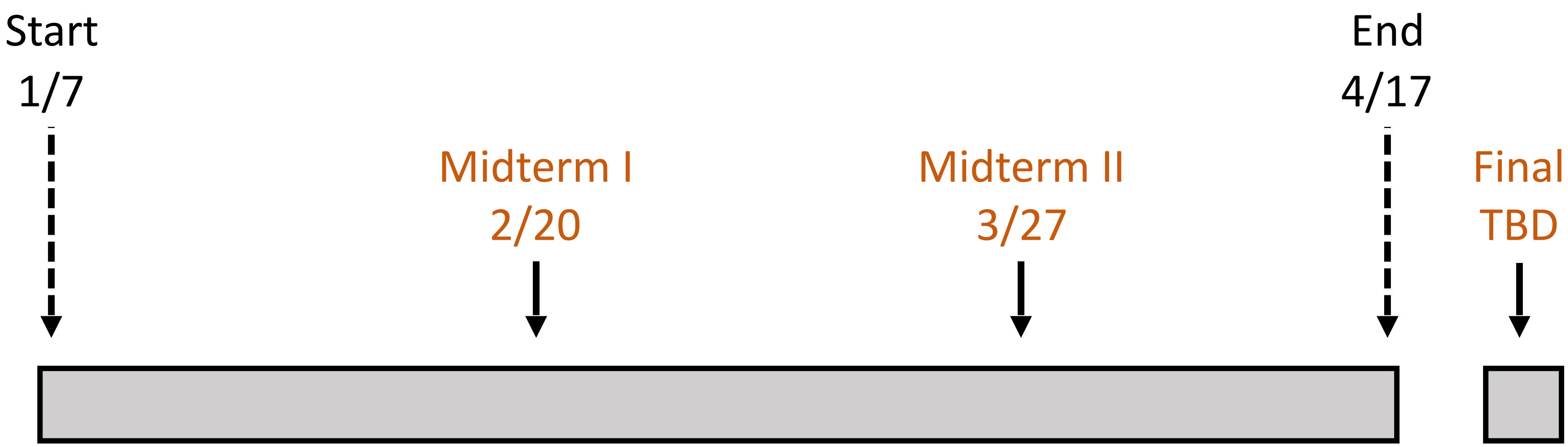
- Improve Communication:
  - How to explain solutions?
  - How to convince someone that a solution is correct?
  - How to convince someone that a solution is best?

Algorithms

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

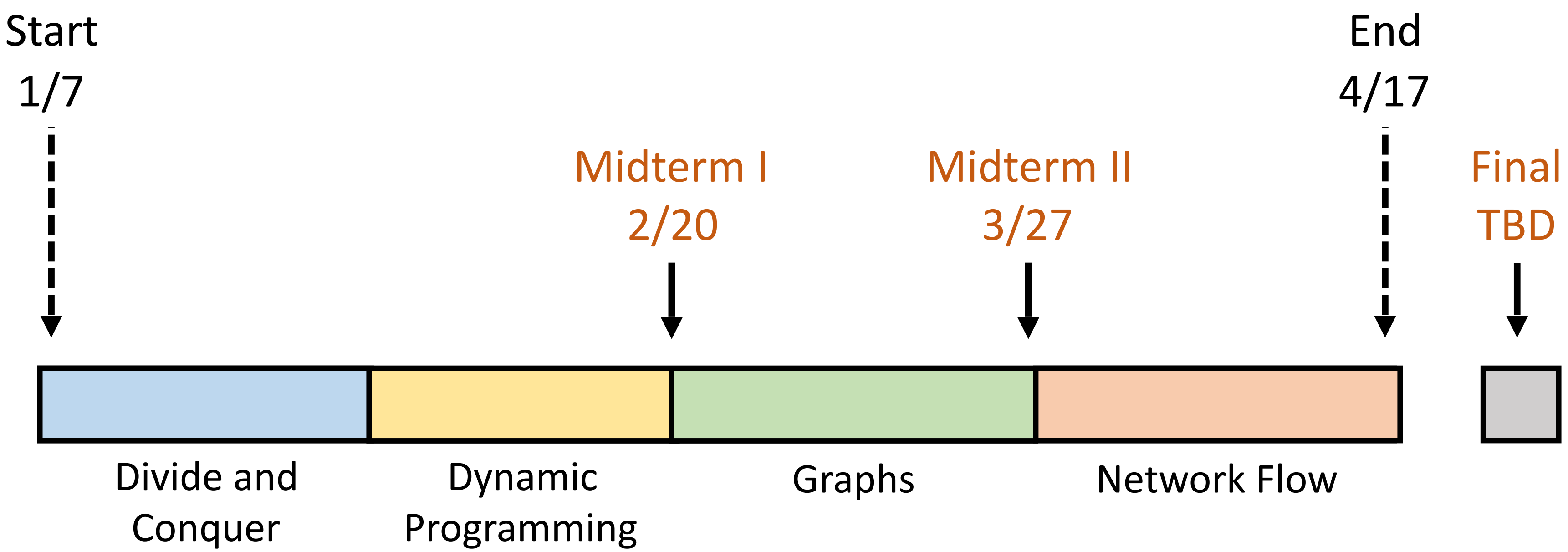
- Get Rich:
  - Many of the world's most successful companies (eg. Google) began with algorithms.
  - Many job interviews have algorithm questions
- Understand the natural world:
  - Brains, cells, networks, etc. often viewed as algorithms.
- Fun:
  - Yes, seriously, fun.

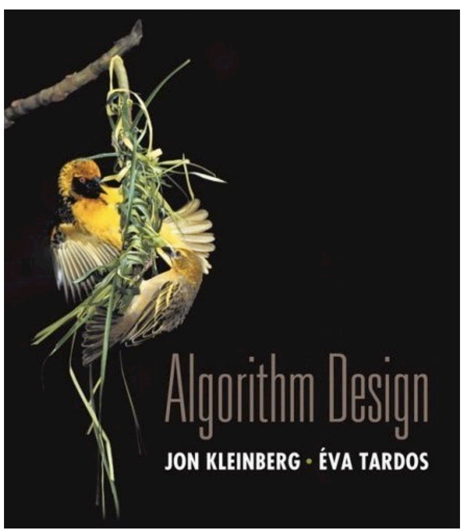
Course Structure



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

Course Structure





Textbook:
Algorithm Design by Kleinberg and Tardos

More resources on the course website

The TA Team

- **TBD**
  - Office Hours: TBD
  - Location: TBD

- **TBD**
  - Office Hours: TBD
  - Location: TBD

- **TBD**
  - Office Hours: TBD
  - Location: TBD

Homework

- Weekly HW Assignments (45% of grade)
  - Due Wednesdays by 2:50pm
  - **HW1 out on Wednesday! Due Wed 1/16**
  - 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

The Not So Short
Introduction to LaTeX 2_ε

Or LaTeX 2_ε 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!
  - More details on Wednesday

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

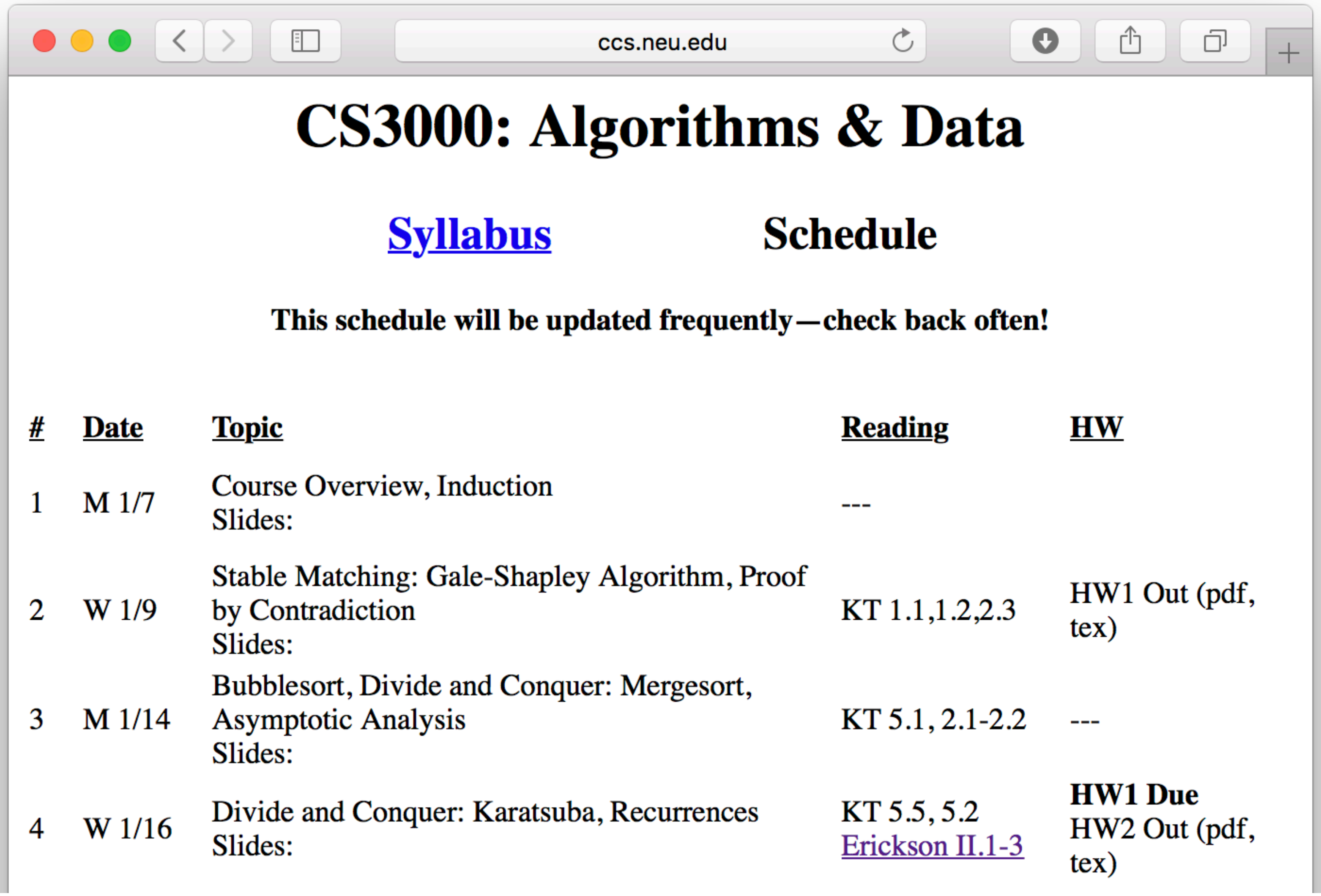
- **Collaboration Policy:**
  - You must write all solutions by yourself
  - You may not share any written solutions
  - You must state all of your collaborators
  - We reserve the right to ask you to explain any solution

Discussion Forum

- We will use Piazza for discussions
  - Ask questions and help your classmates
  - Please use private messages sparingly
- More details on Wednesday!

Course Website

http://www.ccs.neu.edu/home/hand/teaching/cs3000-spring-2018/



CS3000: Algorithms & Data

Syllabus Schedule

This schedule will be updated frequently—check back often!

#	Date	Topic	Reading	HW
1	M 1/7	Course Overview, Induction Slides:	---	
2	W 1/9	Stable Matching: Gale-Shapley Algorithm, Proof by Contradiction Slides:	KT 1.1,1.2,2.3	HW1 Out (pdf, tex)
3	M 1/14	Bubblesort, Divide and Conquer: Mergesort, Asymptotic Analysis Slides:	KT 5.1, 2.1-2.2	---
4	W 1/16	Divide and Conquer: Karatsuba, Recurrences Slides:	KT 5.5, 5.2 Erickson II.1-3	**HW1 Due** HW2 Out (pdf, tex)

What About the Other Sections?

- Prof. Schnyder teaches another section
  - No formal relationship with my section
  - Will cover very similar topics and share some materials
  - Will be out of sync
  - You should not go to OH for Prof. Schnyder's TAs

Illustration:
Let's count how many students are in this class

Simple Counting

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

Recursive Counting - Divide and Conquer

```
RecursiveCount:
    If you are the only person in group:
        return 1
    Else:
        Split your group into two subgroups of similar size
        (one includes you)
        Appoint a leader of the other subgroup
        Ask that leader how many are in that subgroup
        Determine how many are in your subgroup.
        return # in your subgroup + # in other subgroup
```

Recursive Counting - Divide and Conquer

```
RecursiveCount:
    If you are the only person in group:
        return 1
    Else:
        Split your group into two subgroups of similar size
        (one includes you)
        Appoint a leader of the other subgroup
        Ask that leader how many are in that subgroup
        Determine how many are in your subgroup.
        return # in your subgroup + # in other subgroup
```

- How long does this take with $n=2^m$ students?

Recursive Counting

$T(2^m)$ = # steps that recursive count takes w 2^m ppl

$m=0$: $T(2^0)=1$

$m=1$: $T(2^1)=3+T(2^0)=3+1$

- 3: · split group up · appoint leader · add together sizes
- $T(2^0)$: wait to find size of each subgroup

$m=2$: $T(2^2)=3+T(2^1)=3+3+1$

$m=3$: $T(2^3)=3+T(2^2)=3+3+3+1$

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

Proof by induction:

You want to prove a statement $H(m)$ is true for all $m = 0, 1, 2, 3, \ldots$

Steps:

- Prove base case. Show $H(0)$ is true.
- Inductive Step.
  Assume $H(m)$. Show $H(m+1)$ is true.
  ("inductive hypothesis")

Running Time - Proof by Induction

- **Claim:** For every number of students $n = 2^m$

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

this is $H(m)$

Claim: Recursive Count takes $3m+1$ steps for an input $n = 2^m$. That is $T(2^m) = 3m+1$

Know $T(2^0) = 1$ & $T(2^m) = 3 + T(2^{m-1})$

Proof:

Base case
$T(2^0) = 1 = 3 \cdot 0 + 1$ ✓

Inductive Case

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

$= 3 + 3m + 1$ (by inductive hypothesis)

$= 3(m+1) + 1$ ✓

- In terms ot n, $T(n) \approx 3 \log_2 n + 1$