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

• 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

• 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 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?
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?

• **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!
Course Structure

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

Typical Grade Distribution

- A, 30%
- B, 50%
- C, 20%
Course Structure

Start 9/7

<table>
<thead>
<tr>
<th>Divide and Conquer</th>
<th>Dynamic Programming</th>
<th>Graphs</th>
<th>Network Flow</th>
<th>Greedy</th>
<th>Misc</th>
</tr>
</thead>
</table>

Midterm I 10/16

Midterm II 11/16

End 12/4

Final TBD

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

The Not So Short
Introduction to LaTeX 2ε

Or BeX 2ε 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!
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/cs3000f18/syllabus.html
http://www.ccs.neu.edu/home/jullman/cs3000f18/schedule.html

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

41 students
22.38 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?
- How long does this take with $n$ students?

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

$= 4\sqrt{n}$

$4\sqrt{n}$ beats $2n$ unless $n$ is tiny.
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

• Is this correct? Why?

Loop Invariant: After each iteration, the sum of the #s of all people 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

\( n = 2^m \)  

- How long does this take with \( n \) students?

\[
T(n) = \begin{cases} 
2 & \text{if } n = 1 \\
2 + T(2^{m-1}) & \text{otherwise}
\end{cases}
\]

\( T(1) = 3 \)
Running Time

\[ T(2^m) = 2 + T(2^{m-1}) \]

- Recurrence: \( T(1) = 3, T(n) = 2 + T([n/2]) \)

\[
\begin{align*}
T(1) &= 3 \\
T(2) &= 2 + T(1) = 2 + 3 \\
T(4) &= 2 + T(2) = 2 + 2 + 3 \\
&\vdots \\
T(2^m) &= 2 + 2 + 2 + \ldots + 2 + 3 = 2^m + 3 \\
T(n) &= 2 \log_2(n) + 3
\end{align*}
\]
Running Time

• **Claim:** For every number of students $n = 2^m$
  
  $T(2^m) = 2m + 3$
Proof by Induction

\[ T(2^m) = 2 + T(2^{m-1}) \]
\[ T(1) = 3 \]

• **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
    \[ T(2^m) = 2m + 3 \]
    \( \forall m \in \mathbb{N} \quad H(m) \) is true \( \implies \) Conclusion

• **Base Case:** \( H(1) \) is true \( \checkmark \)
• **Inductive Step:** For every \( m \geq 1, H(m) \implies H(m + 1) \)
• **Conclusion:** statement is true for every \( m \)
  \[ H(1) \implies H(2) \implies H(3) \implies \ldots \implies H(m) \]
Proof by Induction

\[ T(2^m) = 2 + T(2^{m-1}) \]
\[ T(1) = 3 \]

**Claim:** For every number of students \( n = 2^m \),
\[ T(2^m) = 2m + 3 \]

**IH:** \( H(m) \) is \( T(2^m) = 2m + 3 \)

**BC:** \( H(1) \) is true b/c \( T(2) = 2 + T(1) = 2 + 3 \)

**IS:** Fix any \( m \in \mathbb{N} \). To show \( H(m) \Rightarrow H(m+1) \)

\[ T(2^{m+1}) = 2 + T(2^m) \]
\[ = 2 + (2m + 3) \]
\[ = 2(m+1) + 3 \quad \therefore \quad H(m+1) \text{ is true} \]
Ask the Audience

Who Wants to be a Millionaire?

In the famous painting “Washington Crossing the Delaware,” what future U.S. President is depicted holding the flag?

A: John Quincy Adams  B: Thomas Jefferson  C: James Monroe  D: James Madison
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
- **Recursive counting:** \( T(n) = 2 \log_2 n + 3 \) steps

But for this class, simple counting was faster???
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 $n \to \infty$