Welcome back! Hope you had a great winter break!
Learning goals
Introductions
Who taught your CSE 21 class?

A. Janine Tiefenbruck  
B. Russell Impagliazzo  
C. Miles Jones  
D. Mia Minnes  
E. Other/Not Applicable

To change your remote frequency:

1. Press and hold power button until flashing
2. Enter two-letter code CA
3. Checkmark / green light indicates success

Why use clickers?
Logistics

- lecture - group discussions, REEF
- discussion sections
- homework - frequent, individual, graded at random
- exams - 2 midterms, final on Saturday before finals week
- textbook - Sipser, Intro to Theory of Computation, 3rd edition
- Piazza - no homework questions
- Gradescope - regrade budget
- JFLAP - visualization software
- OSD students - contact me ASAP

http://cseweb.ucsd.edu/classes/wi17/cse105-ab/
## Logistics

<table>
<thead>
<tr>
<th>Grading</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exams</strong></td>
<td>75%</td>
</tr>
<tr>
<td>MAX ( (Final 35%, Exam 1 20%, Exam 2 20%), (Final 50%, Best Exam 25%))</td>
<td></td>
</tr>
<tr>
<td><strong>Homework</strong></td>
<td>20%</td>
</tr>
<tr>
<td>Number of homework points earned divided by number of homework points possible (after dropping two lowest scores)</td>
<td></td>
</tr>
<tr>
<td><strong>Class Participation</strong></td>
<td>5%</td>
</tr>
<tr>
<td>MAX( Number of participation points earned, 25) divided by 25</td>
<td></td>
</tr>
</tbody>
</table>

http://cseweb.ucsd.edu/classes/wi17/cse105-ab/
How to excel

• Prepare ahead of class
  • Read assigned textbook sections
• Engage in class
  • Discuss questions with your neighbors, look for (counter)examples
  • Go over wrong choices too!
• Reinforce after class
  • Briefly summarize what you learned
  • Start on homework right away
• Seek help and seek to help others, with integrity
How to excel with integrity

It's an integrity violation to…
• Click in for someone who is absent or otherwise falsify attendance
• Ask others to give you HW or exam answers
• Share written answers on HW
• Discuss exam contents with anyone who hasn’t taken the exam
• Search the internet or other resources not provided to the class for HW solutions
• Continue writing on an exam after time is called.

This not a complete list … you are responsible for knowing and following the guidelines. Academic integrity violations will be taken seriously and reported immediately.
Questions about computers

Can they solve problems?

Can algorithms be provably correct?

Can algorithms be made more efficient?

Can algorithms always be found?
Questions about computers

Can they solve problems?

implementation: Java, C++, Python …

Can algorithms be provably correct?

CSE 20, 21, 101

Can algorithms be made more efficient?

Can algorithms always be found?
Questions? Problems?

• Scheduling
• Sorting
• Classifying
• Computing a value
• Predicting

Making a decision or computing a value based on some input
Questions about computers

Can they solve problems?

implementation: Java, C++, Python ...

Can algorithms be provably correct?

CSE 20, 21, 101

Can algorithms be made more efficient?

Can algorithms always be found?

are there problems that no computer can solve?
Questions about computers

Can they solve problems?

implementation: Java, C++, Python …

Can algorithms be provably correct?

CSE 20, 21, 101

Can algorithms be made more efficient?

Complexity theory  [Last week of CSE 105]

Can algorithms always be found?

Computability theory  [Last third of CSE 105]
But what is an algorithm?

- Computers are everywhere in various forms
  - Mobile,
  - Embedded,
  - High-performance,
  - Hardware,
  - Cryptography,
  - Big data
  - DNA/RNA computing
  - Quantum

Different contexts call for different algorithms + different performance constraints
Any common thread?

• Model of computation
  • Abstraction
  • Isolate common features of computation
  • May be removed from implementation

• Why?
  • Wide application
  • Can study limits
Models of computation

- Finite automata and their variants
- Pushdown automata
- Turing machines
Automata

- Text processing
  grep, regexp

- Hardware design
  Moore machines, Mealy machines

- Controllers / Robots

- Statistical models

Code input as strings
Model memory using states
Example: subway turnstile

• A subway turnstile is locked until a token is entered, at which point it unlocks in response to a single push, after which it locks again.

When you approach turnstile, will it open?

How can we model this problem?
Example: subway turnstile

- A subway turnstile is locked until a token is entered, at which point it unlocks in response to a single push, after which it locks again.

When you approach turnstile, will it open?

How can we model this problem?
Inputs: {token entered, push}
Example: subway turnstile

- A subway turnstile is locked until a token is entered, at which point it unlocks in response to a single push, after which it locks again.
Example: subway turnstile

• A subway turnstile is locked until a token is entered, at which point it unlocks in response to a single push, after which it locks again.

What happens if the turnstile is pushed while locked?

A. Transition to unlocked
B. Stay locked
C. Send error message
D. None of the above
Example: subway turnstile

- A subway turnstile is locked until a token is entered, at which point it unlocks in response to a single push, after which it locks again.
Example: subway turnstile

- A subway turnstile is locked until a token is entered, at which point it unlocks in response to a single push, after which it locks again.

What's the initial state of the turnstile?

A. Locked
B. Unlocked
Example: subway turnstile

- A subway turnstile is locked until a token is entered, at which point it unlocks in response to a single push, after which it locks again.
Finite state machine

Capture patterns in behavior based on (limited) knowledge of what has happened in the past, and current input.

Abstract away details ….
- Input: an element from a finite set of symbols
- “Past”: string of input symbols
Deterministic finite automaton (DFA)

- Input: finite string over a fixed alphabet
- Output: "accept" or "reject"

Language of machine is set of strings it accepts (recognizes)
Deterministic finite automaton (DFA)

- Input: finite string over a fixed alphabet
- Output: "accept" or "reject"

Computation of the machine on an input string?

Each new symbol of input gives information about the string.
Deterministic finite automaton (DFA)

- Input: finite string over a fixed alphabet
- Output: "accept" or "reject"

Example input: 0001

Start state (triangle/arrow)

Accept state (double circle)
Deterministic finite automaton (DFA)

- Input: finite string over a fixed alphabet
- Output: "accept" or "reject"

What is the sequence of states followed on input 110?

A. q1, q1, q0  
B. q1, q1, q2  
C. q0, q1, q2, q3  
D. q0, q1, q1, q2  
E. None of the above
Deterministic finite automaton (DFA)
- Input: finite string over a fixed alphabet
- Output: "accept" or "reject"

Does this DFA accept the string 001000?

A. Yes
B. No
C. I don't know
Deterministic finite automaton (DFA)

- Input: finite string over a fixed alphabet
- Output: "accept" or "reject"

Does this DFA accept the empty string?

A. Yes
B. No
C. I don't know