Welcome to CSE21!

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Instructor</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture B</td>
<td>Miles Jones</td>
<td>MWF 9-9:50pm</td>
<td>PCYN 109</td>
</tr>
<tr>
<td>Lecture D</td>
<td>Russell (Impagliazzo)</td>
<td>MWF 4-4:50am</td>
<td>Center 101</td>
</tr>
</tbody>
</table>

http://cseweb.ucsd.edu/classes/sp16/cse21-bd/

March 28, 2016
About this course

Formulate & solve problems

Describe data

Analyze algorithms

Using math
About this course

Why is math part of the CS curriculum?

Proofs: key to convincing arguments, but also key part of software engineering

Vocabulary: basic language of Computer Science

Quantitative Analysis: are our solutions / programs / algorithms good enough? How much computational resources (time, memory, power) does our solution use?
About you

PCYN 109: ??    Center 101: ??

To change your remote frequency
1. Press and hold power button until flashing
2. Enter two-letter code
3. Checkmark / green light indicates success

Have you used iClickers before?

A. Yes
B. No
About you

PCYN 109: ??     Center 101: ??

To change your remote frequency
1. Press and hold power button until flashing
2. Enter two-letter code
3. Checkmark / green light indicates success

Did you take CSE 20 at UC San Diego?

A. Yes
B. No, I took Math 15A instead
C. No, I took an equivalent course
D. No, for some other reason.
About you

PCYN 109: ??    Center 101: ??

To change your remote frequency
1. Press and hold power button until flashing
2. Enter two-letter code
3. Checkmark / green light indicates success

What other CSE class are you taking this quarter?
A. None.
B. CSE 12.
C. CSE 11.
D. CSE 8B.
E. Some other CSE class.
Introductions
What do we assume you know?

Short answer: HW 1.

Longer answer: Rosen Chapters 1, 2, some of 5, some of 9.

Longest answer: You can describe algorithms and their correctness using precise mathematical terminology and techniques. For example:

- Sets, relations (equivalence relations, orders)
- Logical equivalence, conditionals, hypotheses, conditionals, contrapositives
- Universal and existential quantifiers
- Proof by contradiction (indirect proof)
- Proof by induction
- Algorithm invariants
Logistics, part 1

Textbook: Rosen 7th Edition

Participation: Class times (iClicker questions)  https://tritoned.ucsd.edu
   Discussion (quizzes)  https://sections.ucsd.edu

Exams:  First Exam: Friday, April 22
   Second Exam: Friday, May 20
   Final Exam: B00: Wednesday, June 8 (8-11 AM)
          D00: Thursday, June 9 (3-6 PM)
Websites:

**Class Website:** http://cseweb.ucsd.edu/classes/sp16/cse21-bd/
- Homework assignments, calendar, announcements, study guides, contact info, lecture slides (avail. Day after lecture.)

**Gradescope:** gradescope.com
- Homework submission and exam return.

**TritonEd (Ted):** https://tritoned.ucsd.edu
- Participation scores.

**Piazza:** Announcements and Q&A. Contact instructors here! No HW questions on Piazza.

**Office hours:** Instructors and tutors. Discuss HW questions here!
Exams (60%), HW (35%), Participation (5%)

* Details on class website: http://cseweb.ucsd.edu/classes/sp16/cse21-bd/
* Drop lowest HW score
* Drop lowest midterm score if do better on final
* Can use note sheet for exams
* Participation earned via class participation, discussion quizzes, and piazza
* Credit for participation if answer 80% of clicker question in that day's class
* HW and exams answers evaluated not only on the correctness of your answers, but on your ability to effectively communicate your ideas and convince the reader of your conclusions through proofs and logical reasoning.
You’re working on a homework question and run across a definition you don’t understand. You Google the term and the first hit includes a full solution to the homework question. You avoid reading the solution and close the browser. You keep working on the solution and hand in the assignment, without mentioning the Google search since you didn’t use the result. Is this acceptable?

A. Yes  B. No
Academic Integrity Scenarios

You’re not sure if you are interpreting a homework problem correctly. You write a post on Piazza showing what you did to answer it, and asking if this is the correct way of interpreting the question. Is this acceptable?

A. Yes  B. No
Academic Integrity Scenarios

You form a study group with two friends and start working on the next homework. Since there are 6 questions you each pick two questions, think about them, and write out your solutions in a shared Google doc. You glance over each other's work before turning in the assignment. Is this acceptable?

A. Yes       B. No
Goals

1. **Learn concepts** which computer science relies upon:

   Algorithms
   Asymptotic notation
   Recurrence relations
   Graphs
   Enumeration and data representation
   Probability
An example of CS vocabulary: Trees

**Data structure**: Binary search trees

Stay tuned: Chapter 11 in Rosen, Week 6
An example of CS vocabulary: Trees

**Algorithm**: parsing

```plaintext
program
  int id(main) ( ) { code }
  instruction code
    id(cout) ... instruction
    return 0 ;
```
An example of CS vocabulary: Trees

**Model:** possible paths of computation
An example of CS vocabulary: Trees

**Model:** Phylogenetic (evolutionary) tree
**State space**: possible configurations of a game
Conclusion: Many different applications but same underlying idea.

- How do we define a tree?
- What properties are guaranteed by this definition?
- What algorithms can exploit these properties?
2. Solve problems.

- Come up with *new* algorithms

- Think of the homework questions as puzzles that you need to unravel: the solution or even the approach won't be clear right away.

- You can work on homework in groups of 1-3 students.
* Assume elements of the set to be sorted have some underlying order
Which of the following collections of elements is listed in sorted order?

A. 42, 10, 30, 25
B. 10, 25, 30, 40
C. 40, 30, 25, 10
D. All of the above
E. None of the above
Why sort?

A TA facing a stack of exams needs to input all 400 scores into a spreadsheet where the students are listed in alphabetical order.

OR

You want to find all the duplicate values in a long list.
A TA facing a stack of exams needs to input all 400 scores into a spreadsheet where the students are listed in alphabetical order.

OR

You want to find all the duplicate values in a long list.

It's easier to access data when it is sorted because you know exactly where to find it.
DIY: Sorting Algorithms

1. **Find a group** of about 20 people nearby. Write your first names on separate papers.

2. **Sort** the names of the people in your group alphabetically by first name.

3. **Discuss as a group** the strategy you used to sort the papers, and how you might describe it to someone else.

4. **Write** a clear English description of the strategy your group used (each person should do this.)

5. **Select** one representative to describe your group's strategy on the board.
Discussion of Sorting Algorithms

Is the strategy clear?

Will the strategy always work?

Does the strategy scale well to bigger groups?
General questions to ask about algorithms

1) **What** problem are we solving?
2) **How** do we solve the problem?
3) **Why** do these steps solve the problem?
4) **When** do we get an answer?
General questions to ask about algorithms

1) **What** problem are we solving?  
   **PROBLEM SPECIFICATION**

2) **How** do we solve the problem?  
   **ALGORITHM DESCRIPTION**

3) **Why** do these steps solve the problem?  
   **CORRECTNESS**

4) **When** do we get an answer?  
   **RUNNING TIME PERFORMANCE**
Given a list

\[ a_1, a_2, \ldots, a_n \]

rearrange the values so that

\[ a_1 \leq a_2 \leq \ldots \leq a_n \]

Values can be any type (with underlying total order). For simplicity, use integers.
Selection Sort (Min Sort)

"Find the first name alphabetically, move it to the front. Then look for the next one, move it, etc."
procedure selection sort(a1, a2, ..., an: real numbers with n >= 2 )
for i := 1 to n-1
  m := i
  for j:= i+1 to n
    if ( aj < am ) then m := j
  interchange ai and am

{ a1, ..., an is in increasing order}
Bubble Sort

"Compare the first two cards, and if the first is bigger, keep comparing it to the next card in the stack until we find one larger than it. Repeat until the stack is sorted."
procedure bubble sort(a1, a2, ..., an: real numbers with n >=2 )
for i := 1 to n-1
  for j:= 1 to n-i
    if ( aj > aj+1 ) then interchange aj and aj+1

{ a1, ..., an is in increasing order}
"We passed the cards from right to left, each individual inserting their own card in the correct position as they relayed the pile."
procedure insertion sort(a1, a2, ..., an: real numbers with n >=2 )
for j := 2 to n
  i := 1
  while aj > ai
    i := i+1
  m := aj
  for k := 0 to j-i-1
    aj-k := aj-k-1
  ai := m

{ a1, ..., an is in increasing order}
"Call out from A to Z, collecting cards by first letter. If there are more than one with the same first letter, repeat with the second letter, and so on."
Bucket Sort – Pseudo pseudo code

- Create empty buckets that have an ordering.
- Put each of the elements of the list into the correct bucket.
- Sort within each bucket.
- Concatenate the buckets in order.
"We split into two groups and organized each of the groups, then got back together and figured out how to interleave the groups in order."
Merge Sort – Pseudo pseudo code

• If the list has just one element, return.
• Otherwise,
  • Divide list into two pieces:
    • L1 = a1 . . . an/2 and L2 = an/2+1 . . . an
  • M1 = Merge sort ( L1 )
  • M2 = Merge sort ( L2 )
  • Merge the two (sorted) lists M1 and M2

Rosen page 196, 367-370
Others?

- Bogo sort
- Quick sort
- Binary search tree traversal

Why so many algorithms?
Why so many algorithms?

Practice for homework / exam / job interviews.

Some algorithms are better than others. Wait, better?
Reminders

- Read syllabus on class website.
- Enroll in Piazza and Gradescope.
- Register iClicker.
- Sign up for discussion section. Discussion sections start today.

HW 1 due **Wednesday 11:59pm.**