Welcome to CSE21!

	Miles Jones	**MTThF 8:30-9:50pm**	**CSE 4258**

August 1, 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

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

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

Where are you from?

A. San Diego
B. California
C. United States
D. Other country

About you

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

Where do you think I am from?

A. California
B. Virginia
C. Maine
D. Texas
E. Florida

About you

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 in summer academy?

A. None.
B. CSE 12
C. CSE 15L

Introductions

What do we assume you know?

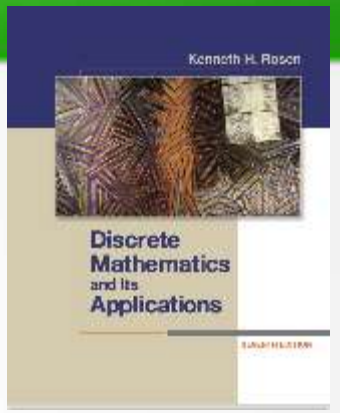
Short answer: HW 1.

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
- Logarithm rules.

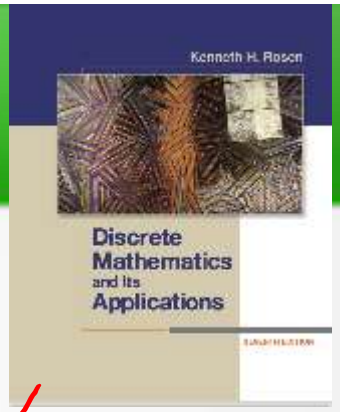
Logistics



Textbook: Rosen 7th Edition

Exams: **First Exam:** Friday, August 12
Second Exam: Friday, August 26
Final Exam: Friday, September 2

Logistics, part 2



Websites: http://sahilag.github.io/cse 21/

Class Website:

Homework assigments, calendar, announcements, study guides, contact info, lecture slides (avail. Day after lecture.)

Gradescope: gradescope.com

Homework submission and exam return.

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

Office hours: Instructors and tutors. Discuss HW questions here!

Logistics, part 2

8am						
9am		8:30 – 9:50 Lecture cse 4258	8:30 – 9:50 Lecture cse 4258	8:30 – 9:30 Study Session	8:30 – 9:50 Lecture cse 4258	8:30 – 9:50 Lecture cse 4258
10am		10 – 10:50 Discussion - Sahil	10 – 10:50 Problem Solving - Soheil, Julia, Tommy, Scarlet		10 – 10:50 Discussion- Annie	10 – 10:50 Problem Solving - Julia, Sahil, Scarlet, Soheil
11am		11 – 11:50 Office Hours - Sahil soheil miles	11 – 11:50 Office Hours - Annie		11 – 11:50 Office Hours - Miles, Julia, Soheil	11 – 11:50 Office Hours - Sahil
12pm						
1pm						
2pm						
3pm						
4pm						
5pm		5p – 6p Staff Meeting in Miles Miles Office				
6pm						
7pm	7p – 9p Office Hours - Scarlet, Tommy	7p – 9p Office Hours - Annie, Tommy, Scarlet, Julia	7p – 9p Office Hours - Tommy, Julia, Scarlet	7p – 9p Office Hours - Tommy	7p – 9p Office Hours- Sahil, Julia, Scarlet, Tommy	
8pm						
9pm						

Logistics, part 2

- Your grade will be based on a combination of exams (70%) and homework (30%)
- Your lowest homework will be dropped and your lowest midterm exam may be dropped if you do better on the final.
- Your work will be evaluated not only on the correctness of your answers but on your ability to effectively communicate your ideas.

Academic Integrity Scenarios

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

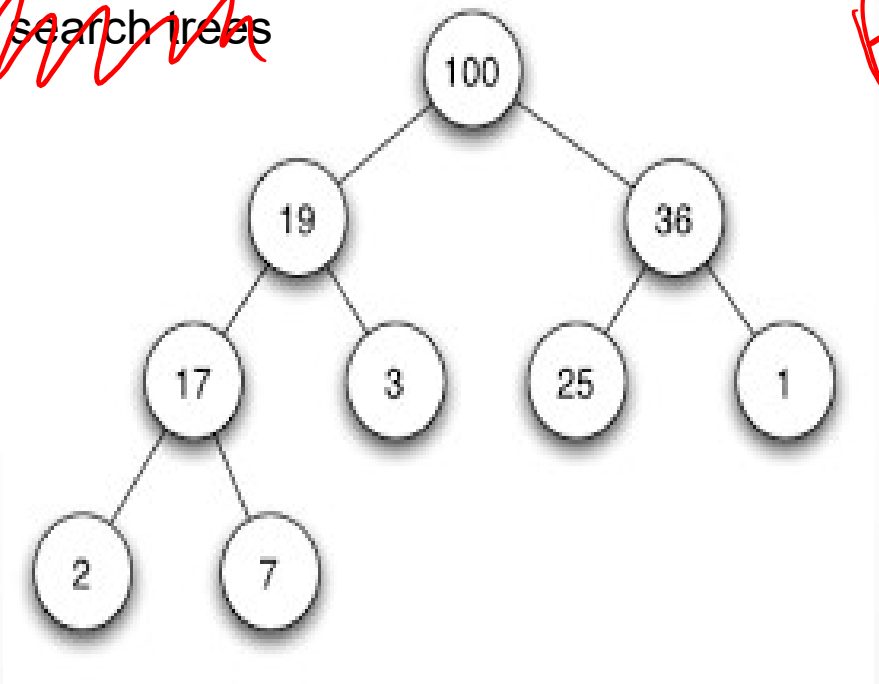
Goals

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

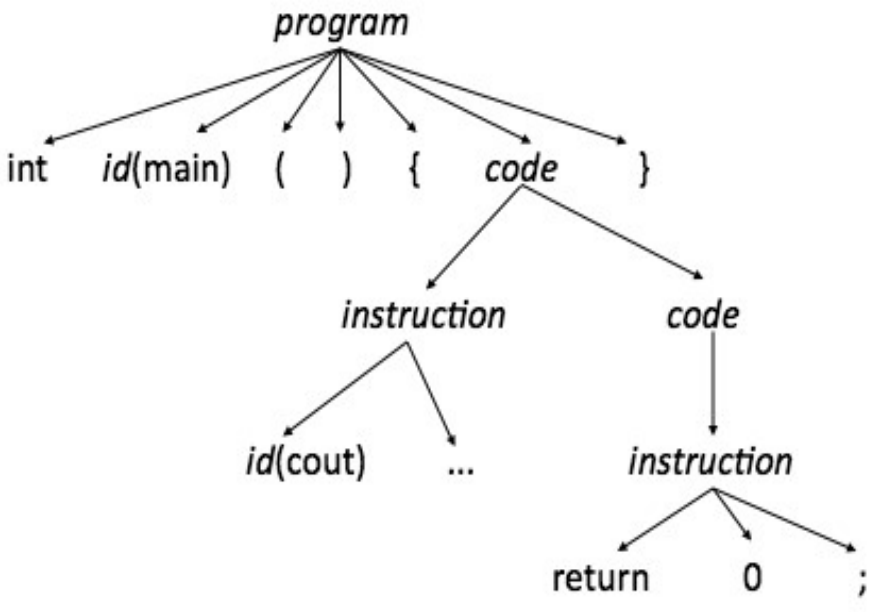


Binary tree.

Stay tuned: Chapter 11 in Rosen, Week 6

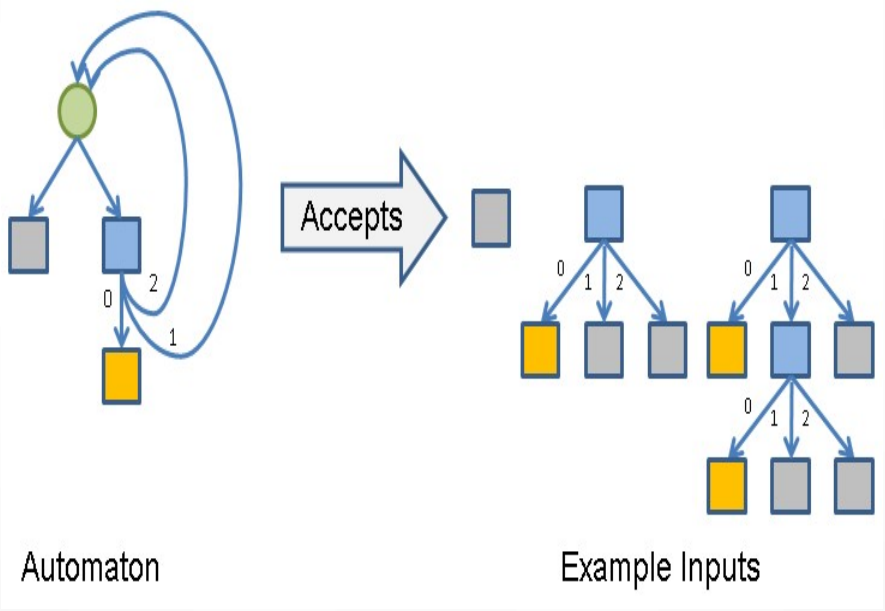
An example of CS vocabulary: Trees

Algorithm: parsing



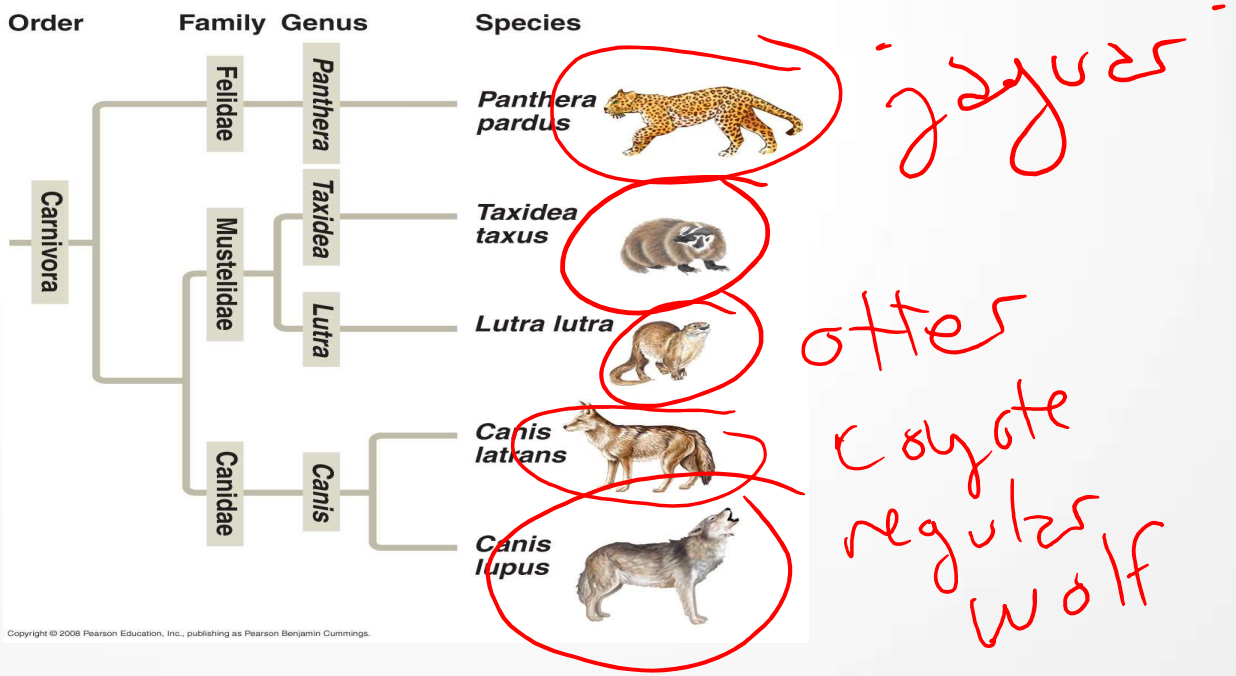
An example of CS vocabulary: Trees

Model: possible paths of computation



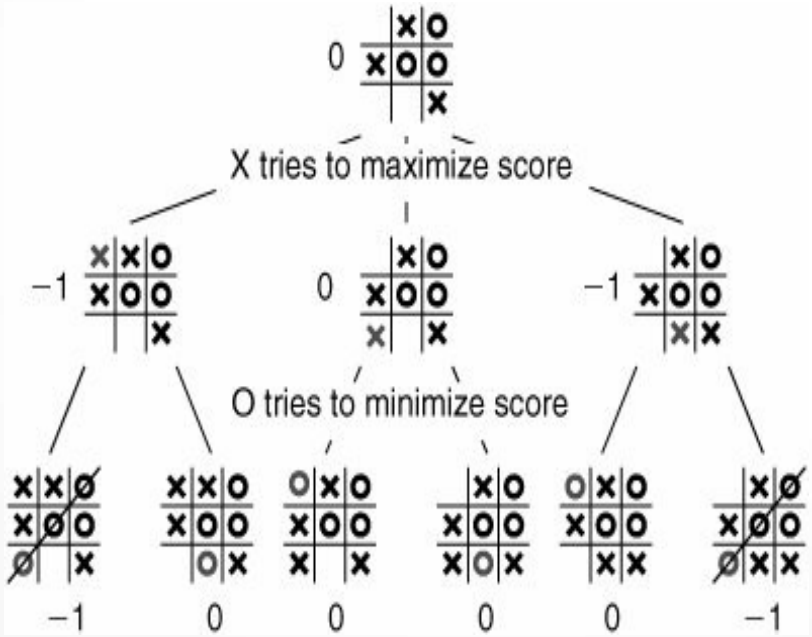
An example of CS vocabulary: Trees

Model: Phylogenetic (evolutionary) tree



An example of CS vocabulary: Trees

State space: possible configurations of a game



An example of CS vocabulary: Trees

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?

Goals

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-~~4~~ 2 students.

Sorting (or Ordering)

Section 3.1 in Rosen

vs.

* Assume elements of the set to be sorted have some underlying order

Sorting (or Ordering)

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.

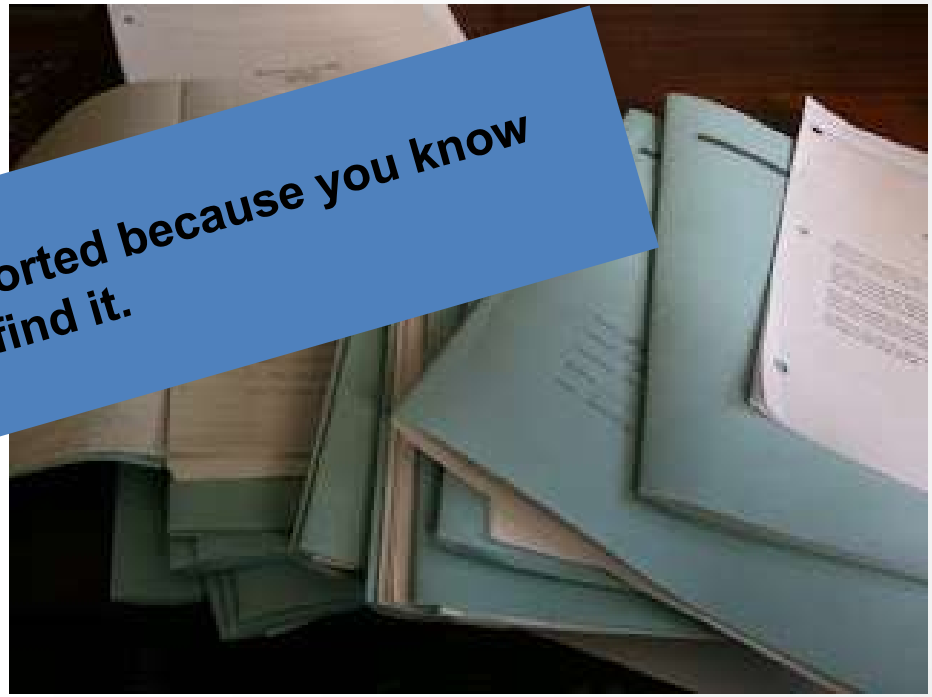
Why sort?

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
value

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** we will divide the class in half

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.

https://youtu.be/ywWBy6J5gz8 Sorting Dance.

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? CORRECTNESSS

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

Sorting: Specification: WHAT

Rosen page 196

Given a list

$$a_1, a_2, \ldots, a_n$$

rearrange the values so that

$$a_1 <= a_2 <= \ldots <= a_n$$

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

Your approaches: HOW

Selection Sort (Min Sort)

"Find the first name alphabetically, move it to the front. Then look for the next one, move it, etc."

Selection Sort (MinSort) Pseudocode

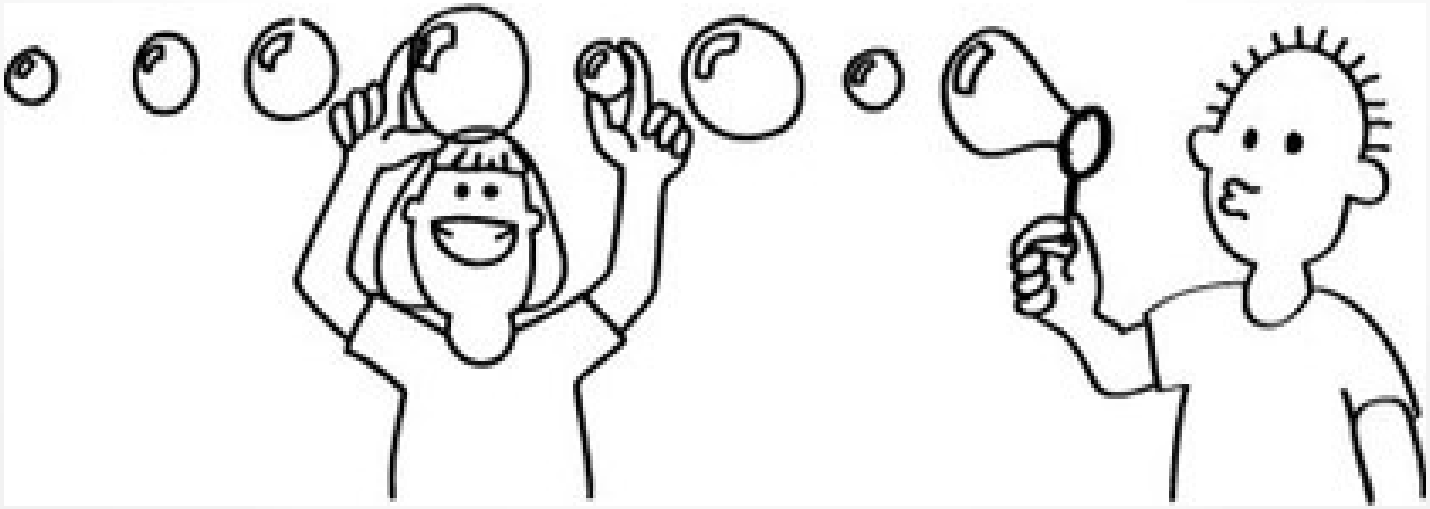
Rosen page 203, exercises 41-42

```
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."



Bubble Sort Pseudocode

Rosen page 197

```
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}
```

Insertion Sort

"We passed the cards from right to left, each individual inserting their own card in the correct position as they relayed the pile."

Insertion Sort Pseudocode

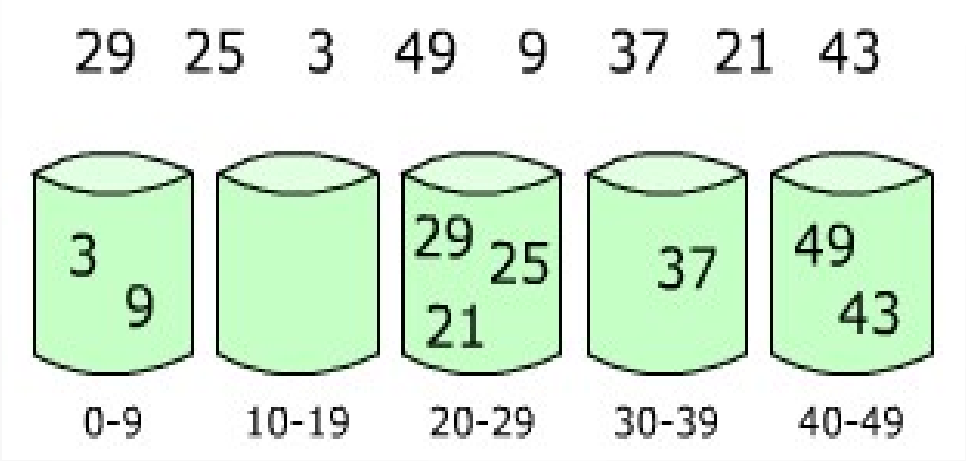
Rosen page 198

```
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}
```

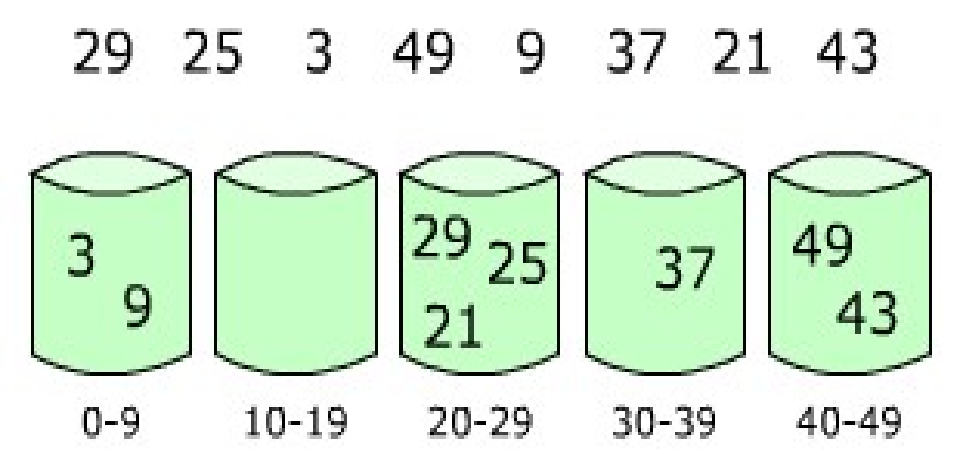
Bucket Sort

"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.



Merge Sort

"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

Rosen page 196, 367-370

- If the list has just one element, return.
- Otherwise,
  - Divide list into two pieces:
    $L_1 = a_1 \ldots a_{n/2}$ and $L_2 = a_{n/2+1} \ldots a_n$
  - M_1 = Merge sort (L_1)
  - M_2 = Merge sort (L_2)
  - Merge the two (sorted) lists M_1 and M_2

Others?

Bogo sort

Quick sort

Binary search tree traversal

https://en.wikipedia.org/wiki/Sorting_algorithm

Why so many algorithms?

Why so many algorithms?

Practice for homework / exam / job interviews.

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

Reminders

HW 1 Due Tuesday August 2 at 11:59pm on gradescope.