CSSE 230 Data Structures and Algorithm Analysis
Day 1
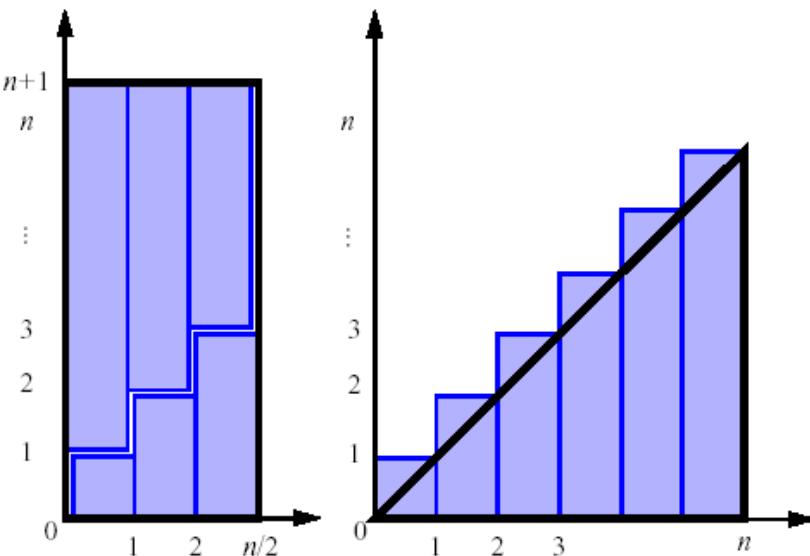
Brief Course Intro
Math Review
Growable Array Analysis

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \ldots + n = \frac{n^2 + n}{2}$$

- two visual representations
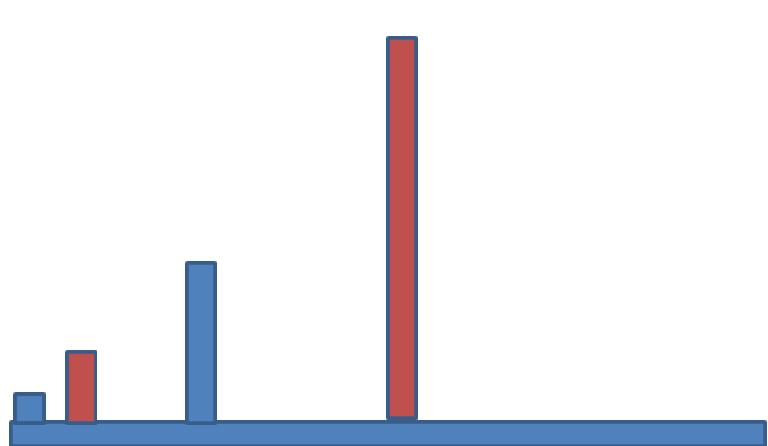
Introductions

▸ Roll call
  - ◦ Introduce yourself to the person next to you
  - ◦ You'll share more with classmates on discussion forum, like what work you've done that you are most proud of.

▸ Dr. Chenette
  - ◦ At R-H since 2014. Math 2014-16, CSSE since 2016.
  - ◦ B.S., Harvey Mudd in Math-CS
  - ◦ Ph.D., Georgia Tech in Algorithms, Combinatorics, & Optimization
  - ◦ Special interests in cryptography, algorithms, discrete math
  - ◦ Courses taught at Rose:
    - • CSSE230, Design & Analysis of Algorithms, Cryptography
    - • DisCo 1 & 2, Calc 2 & 3, DE 1
  - ◦ Hobbies: cycling, running, triathlon, classical music, travel

Heather (CHE faculty)

Sylvia (3 yrs)

Adrienne (15 mo)

Goal: independently design, develop and debug software that uses correct, clear, and efficient algorithms and data structures

Prove: An AVL Tree has O(log n) height

Proof: By definition,

$$| \text{height}(T_L) - \text{height}(T_R) | \leq 1$$

...

Topic	I do	You do	You practice	You show off
Analysis	Explain, show, do	Listen, follow, read, quiz	Homework sets	Tests
Programming			Major programs	Tests, project

```
/**
 * A height-balanced binary tree with rank
 * that could be the basis for a text
 * editor.
 * @author Claude Anderson and Matt Boutell.
 */
public class EditTree {
    private Node root;
    private int rotationCount = 0;
    private Node singleLeftRotation(
        Node grandParent, Node parent) {
        // Set parent nodes

...
```

How to succeed in CSSE230

- Work hard
  - Re-do CSSE220 stuff as needed to make sure your foundations (recursion and linked lists) are strong
- Take initiative in learning
  - Read the text, search Javadocs, come for help
- Focus while in this class
- Start early and plan for no all-nighters
  - Two assignments each week: 1 homework set and 1 major program
- Never give or use someone else's answers

Tools

▸ http://www.rose-hulman.edu/class/csse/csse230/201810/Schedule/: schedule, assignments, room #s!

▸ www.piazza.com, not email: homework questions and announcements
  ◦ If you email, I'll usually reply, "Great question! Please post it to Piazza"
  ◦ It should auto-email you whenever there is a post.
  ◦ Posting and answering posts is a factor in citizenship grade

▸ moodle.rose-hulman.edu: gradebook, homework pdf turn-in, peer evaluations, solutions

After today's class, you will be able to...

- analyze runtimes of code snippets by counting instructions.

- explain why arrays need to grow as data is added.

- derive the average and worst case time to insert an item into an array [GrowableArray exercise]

Analysis/Math Review

Notation

- Floor

$$\lfloor x \rfloor = \text{the largest integer} \leq x$$

- Ceiling

$$\lceil x \rceil = \text{the smallest integer} \geq x$$

- `java.lang.Math`, provides the static methods `floor()` and `ceil()`

Summations

- Summations
  - general definition:

$$\sum_{i=s}^{t} f(i) = f(s) + f(s+1) + f(s+2) + \dots + f(t)$$

  - where f is a function, s is the start index, and t is the end index

Geometric progressions: each term is a constant multiple of the previous term

- Geometric progression: $f(i) = a^i$
  - given an integer $n \geq 0$ and a real number $0 < a \neq 1$

$$\sum_{i=0}^{n} a^i = 1 + a + a^2 + \ldots + a^n = \frac{1 - a^{n+1}}{1 - a}$$

Memorize this formula!

  - geometric progressions exhibit exponential growth

Exercise: What is $\displaystyle\sum_{i=2}^{6} 3^i$?

The sum can also be written:
$$\frac{a^{n+1} - 1}{a - 1}$$

This will be useful for today's Growable Arrays exercise!

Arithmetic progressions: constant difference
Most important to us: a difference of 1

- Arithmetic progressions:
  - An example

Memorize this formula!

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \dots + n = \frac{n^2 + n}{2}$$

Exercise: $\sum_{i=21}^{40} i$

Also useful for today's Growable Arrays exercise!

Visual proofs of the summation formula

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \ldots + n = \frac{n^2 + n}{2}$$

- two visual representations
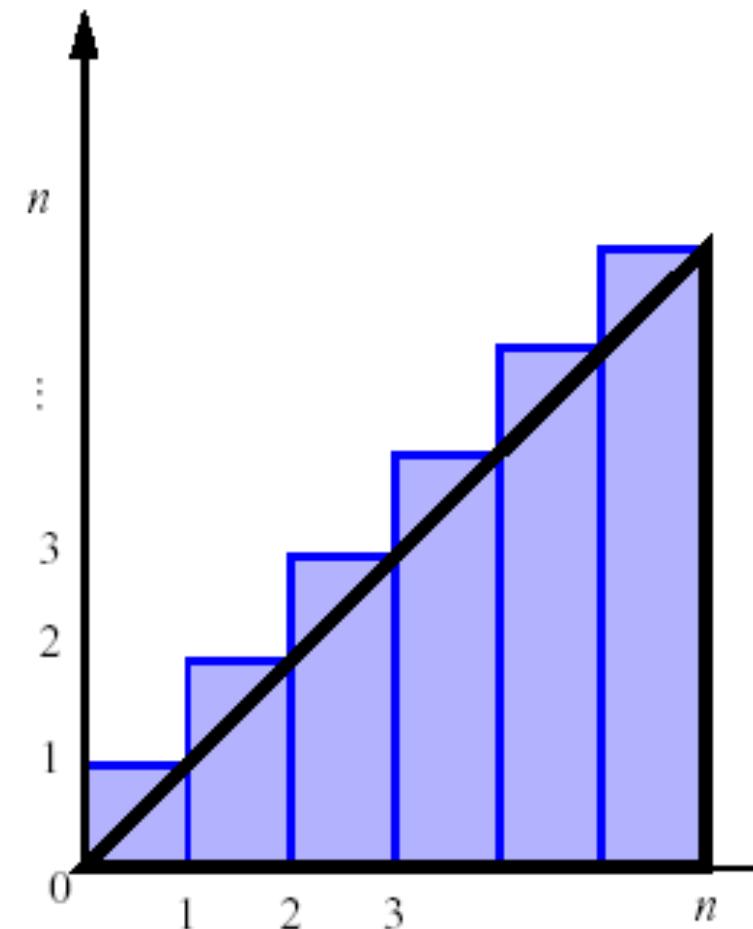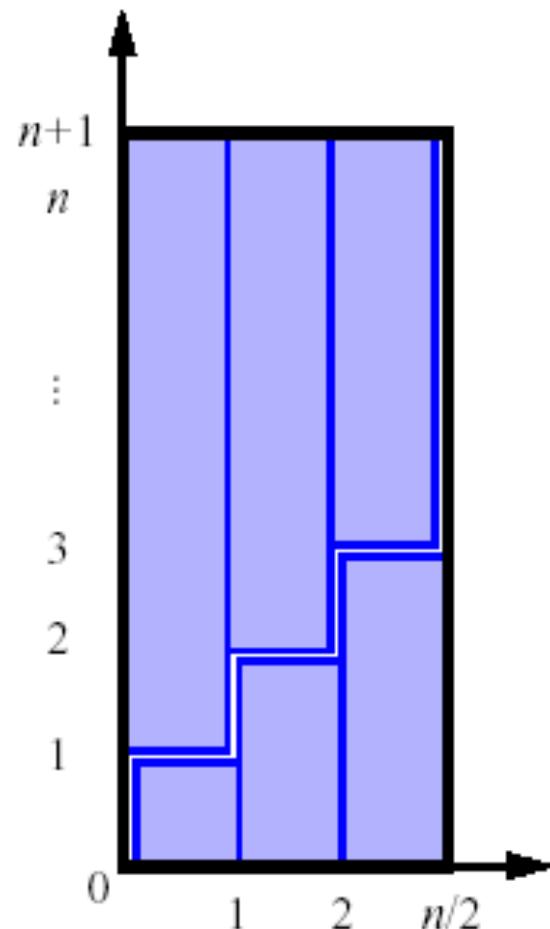
Application: Find exact and big-Oh runtime of Selection Sort

```
for (i=n-1; i>0; i--) {
    int maxPos = 0;
    for (int j = 0; j <=i; j++) {
        if (a[j] > a[maxPos]) {
            maxPos = j;
        }
    }
    swap a[maxPos] with a[i] ;
}
```

- **How many comparisons of array elements are done?**
    - **Exact? Big-Oh?**
- **How many times are array elements copied?**

Growable Array Analysis

An exercise in doubling,
done by pairs of students

Arrays are ubiquitous

▸ Basis for ArrayLists, sorting, and hash tables
▸ Why? O(1) access to any position, regardless of the size of the array.

▸ Limitation:
  ◦ Fixed capacity!
  ◦ If it fills, you need to re-allocate memory and copy items
    • How efficient is this?
    • Consider two schemes: "add 1" and "double"

Growable Arrays

```java
// Read an unlimited number of String; return a String [ ]
public static String [ ] getStrings( ) {
    Scanner in = new Scanner( System.in );
    String [ ] array = new String[ 5 ];
    int itemsRead = 0;
    System.out.println( "Enter any number of strings, one per line; "
    System.out.println( "Terminate with empty line: " );

    while( in.hasNextLine( ) ) {
        String oneLine = in.nextLine( );
        if (oneLine.equals( "" ) )
            break;
        if ( itemsRead == array.length)
            array = resize( array, array.length * 2 );
        array[ itemsRead++ ] = oneLine;
    }

    System.out.println( "Done reading" );
    return resize( array, itemsRead );
}
```

Original array size = 5

We don't know in advance how many strings there will be

Grow when necessary

How does `resize()` work?
What is the main "overhead cost" of resizing?

Work on Growable Array Exercise

- Work with a partner

- Hand in the document before you leave today if possible. Otherwise due start of day 2's class.

- Get help as needed from me and the assistants.