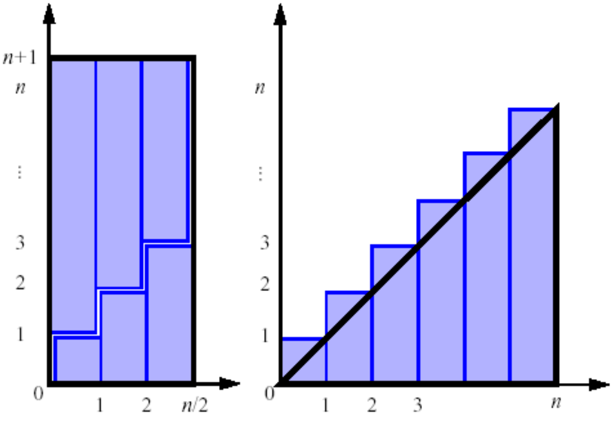
Pick up an in-class quiz from the table near the door

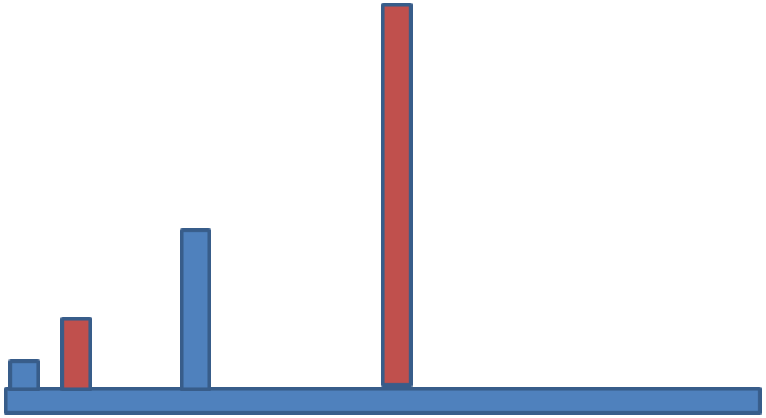
CSSE 230 Data Structures and Algorithm Analysis Day 1

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



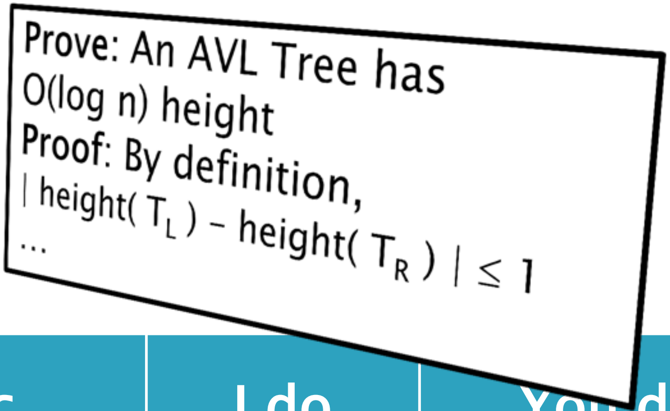


And intro to daily quizzes, worth 5% of grade: Q1

Introductions

- Roll call:
  - Pronunciations and nicknames
  - Where you live on campus
  - You’ll share more with classmates on discussion forum
- Matt:
  - Here since 2005 (but in Zambia in 2011-2012)
  - Taught CSSE120, 120 Robotics, 220, 221, 230, Image Recognition, Android, Cryptography, Fractals, Mechatronics, Robotics senior design

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



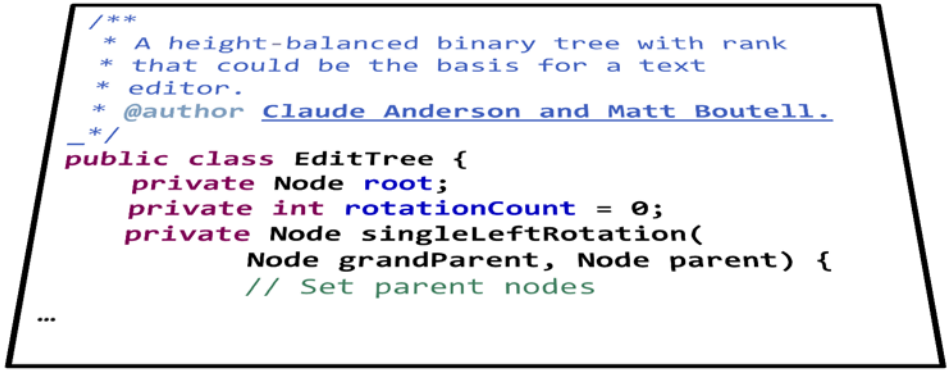
Prove: An AVL Tree has O(log n) height

Proof: By definition,

| height(T_L) – height(T_R) | ≤ 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
...
```

Q2-3

Our expectations

- Recall from the syllabus
  - Work hard
  - 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

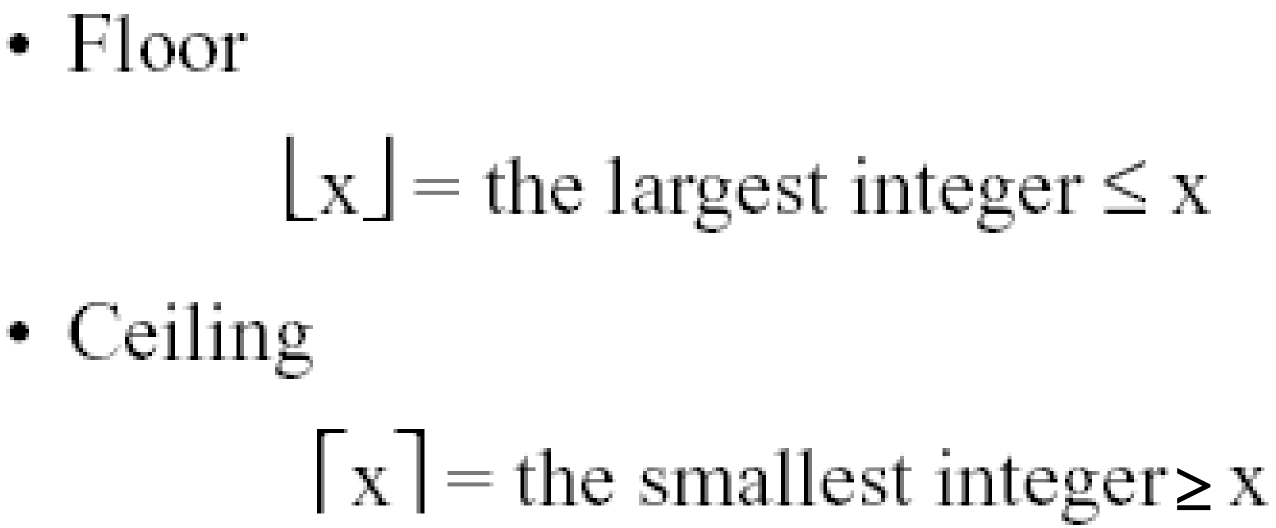
Q4-8

Tools

- http://www.rose-hulman.edu/class/csse/csse230/201420/Schedule/Schedule.htm: schedule, assignments, room #s!
- www.piazza.com, not email: homework questions
  - If you email, we'll usually reply, "Great question! Please post it to Piazza"
- moodle.rose-hulman.edu: gradebook, written assignment pdf turn-in

Analysis/Math Review

Notation



- Floor

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

- Ceiling

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

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

Summations

- Summations
  - general definition:

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

  - 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

Q9-10

- 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 $\sum_{i=2}^{6} 3^i$?

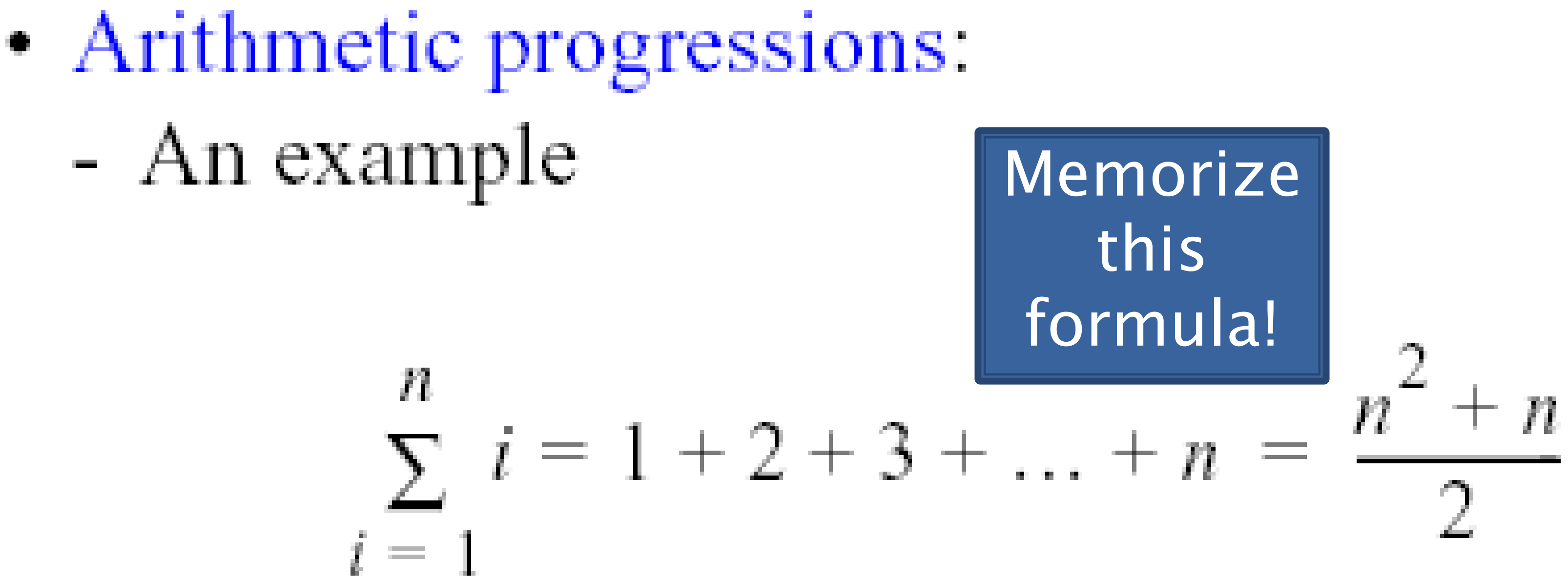
The sum can also be written

$$\frac{a^{n+1} - 1}{a - 1}$$

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

Q11-12

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



- Arithmetic progressions:
  - An example

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

Memorize this formula!

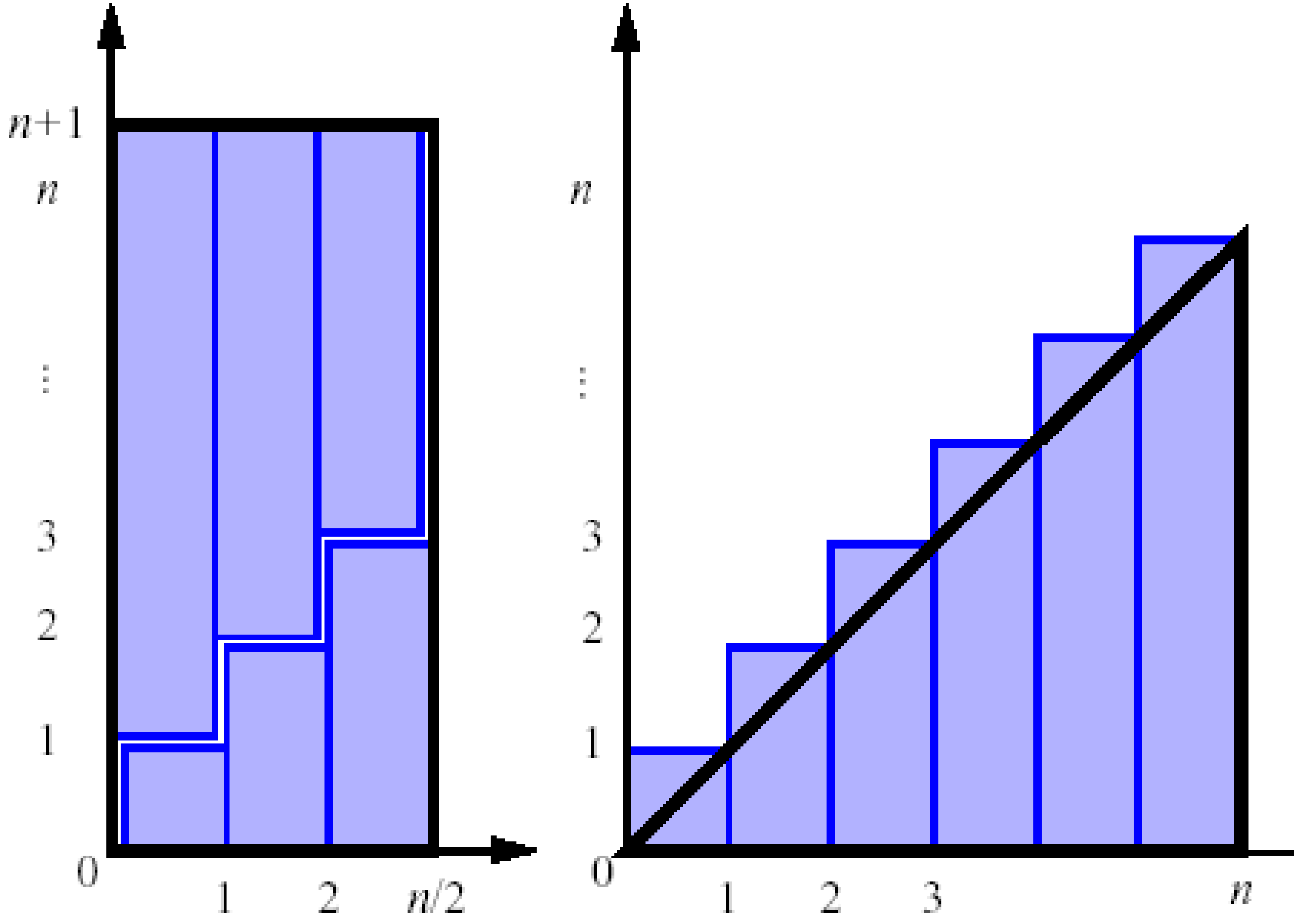
Exercise: $\sum_{i=21}^{40} i$

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

Q13-14, turn in

```
for (i=n-1; i>0; i--) {
    find the largest element among  a[0] ... a[i] ;
    exchange the largest element with a[i] ;
 }
```

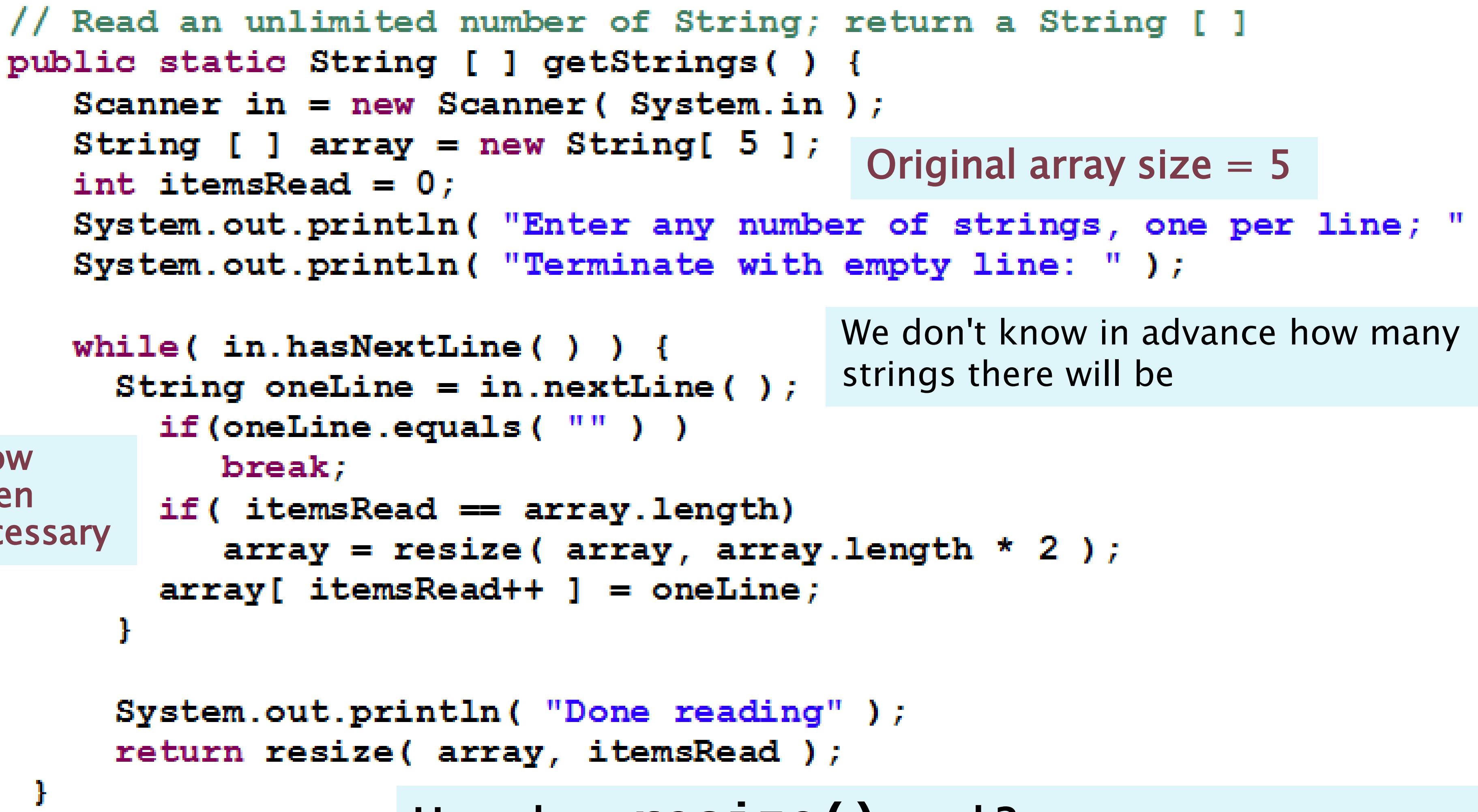
- How many comparisons of array elements are done?
- How many times are array elements copied?

(When you think you have the answers, compare with a partner)

Growable Array Analysis

An exercise in doubling,
done by pairs of students

Growable Arrays



```
// 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
- Get help as needed from me and the assistants.