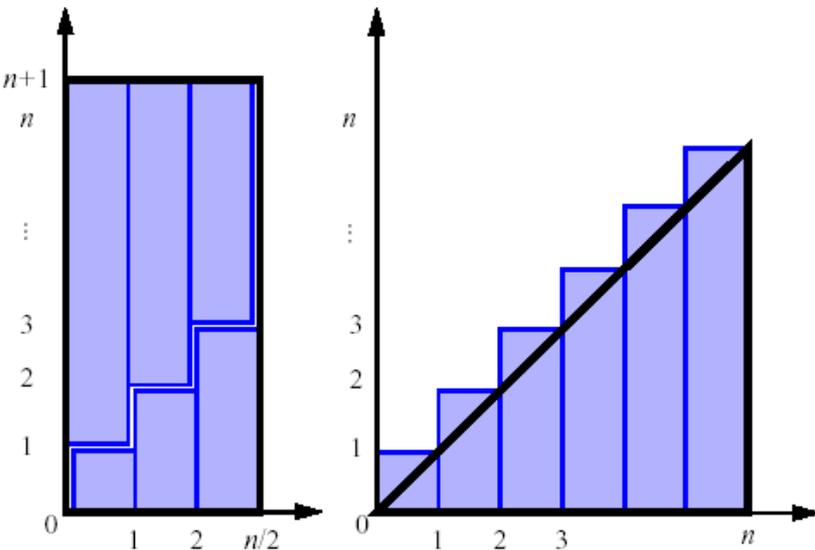


Pick up an in-class quiz from the table near the door

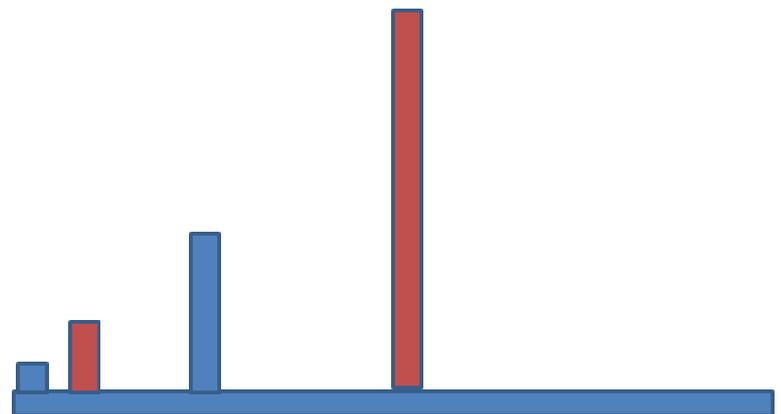
CSSE 230 Data Structures and Algorithm Analysis Day 1

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

- two visual representations



Brief Course Intro
Math Review
Growable Array Analysis



Student Introductions

- ▶ Roll call
 - Introduce yourself to the person next to you
 - Outside of class, please share more with classmates on the Piazza discussion forum, e.g., what's your favorite food, what are your hobbies, types of work you've done, etc.
(5pts toward homework grade)

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 ↓ Programming	Explain, show, do	Listen, follow, read, quiz	Homework sets Major programs	Tests 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  
    }  
    ...  
}
```

Why *efficient* algorithms?

Here's \$1,000,000,000:



- ▶ Find serial number KB46279860I
- ▶ If unsorted, you could look at all 10 million bills.
- ▶ If sorted by serial number, binary search finds it by only looking at _____ bills.

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
 - Search Javadocs, Google, textbook, come for help
- ▶ Focus while in this class
 - [Laptops can distract from learning](#) (11/26/2017 NYT)
- ▶ Start early and plan for no all-nighters
 - Two assignments each week: 1 homework set and 1 major program
- ▶ Talk to and work with others
 - Don't be the "lone ranger"
- ▶ But never give or use someone else's answers

Tools

- ▶ Course webpage
 - Schedule, HW/program assignments, etc.
 - Read the **Syllabus**: Tomorrow's quiz will start with questions about it.
- ▶ Piazza
 - Homework questions.
 - Don't email, use Piazza! If you email, I'll usually reply, "Great question! Please post it to Piazza"
 - You can set it to auto-email you whenever there is a post.
- ▶ Moodle:
 - gradebook, homework pdf turn-in, peer evaluations, solutions
- ▶ Eclipse
 - Demo of checking out WarmUpAndStretching

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: $\lfloor x \rfloor =$ the largest integer $\leq x$
- ▶ Ceiling: $\lceil x \rceil =$ the largest integer $\geq x$
 - `java.lang.Math` provides methods `floor()` and `ceil()`

▶ Summations

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

- f is a function
- s is the start index
- t is the end index

Geometric sums

- ▶ Geometric sequence: each term is a constant multiple of the previous term
 - The sequence exhibits exponential growth
 - $1, a, a^2, a^3, a^4, \dots$

- ▶ Geometric sum. Index is in the *exponent*

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

(for $n \geq 0, a \neq 1$)

Memorize
this
formula!

- ▶ Exercise. Compute

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

Arithmetic sums

- ▶ Arithmetic sequence: each term is a constant (often 1) added to the previous term
 - 2,6,10,14,18, ...
 - 1,2,3,4,5,6, ...

- ▶ Arithmetic sum

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

Memorize
this
formula!

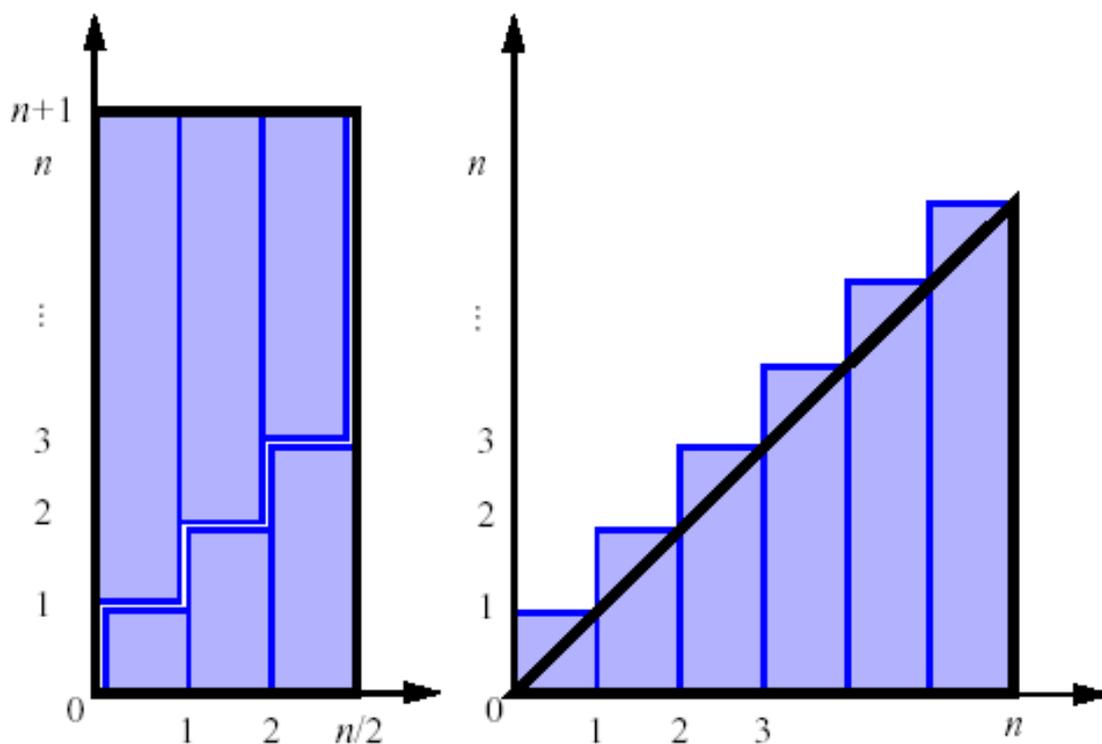
- ▶ Exercise. Compute

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

Visual proof of the arithmetic sum formula

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

- two visual representations



Application: Analysis of Selection Sort

▶ Selection sort basic idea:

- Think of the array as having an **unsorted part**, then a **sorted part**

0	1	2	3	4	5	6	7	8	9
7	388	310	438	79	10	121	537	974	1391

- Find the *largest* value in the **unsorted** part
- Swap it to the *beginning* of the **sorted** part (making the sorted part bigger and the unsorted part smaller)

Repeat until
unsorted part is
empty

▶ Pseudocode:

```
1 for (int i = n-1; i > 0; i--) {
2     int maxPos = 0;
3     for (int j = 0; j <= i; j++) {
4         if (a[j] > a[maxPos]) {
5             maxPos = j;
6         }
7     }
8     swap a[maxPos] with a[i];
9 }
```

Application: Analysis of Selection Sort

```

1 for (int i = n-1; i > 0; i--) {
2     int maxPos = 0;
3     for (int j = 0; j <= i; j++) {
4         if (a[j] > a[maxPos]) {
5             maxPos = j;
6         }
7     }
8     swap a[maxPos] with a[i];
9 }

```

How many times does the most-frequently-run line of code run, as a function of n ?

Tabulate all values of index variables.

i	j	Count
$n-1$	$0,1,2,\dots,n-1$	n
$n-2$	$0,1,2,\dots,n-2$	$n-1$
$n-3$	$0,1,2,\dots,n-3$	$n-2$
...
2	$0,1,2$	3
1	$0,1$	2

Add up counts!

$$\sum_{i=2}^n i = ?$$

Note: not the same i as before...

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 of ArrayLists:
 - 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”
- ▶ GrowableArray demo

Work on Growable Array Exercise

- ▶ Work with a partner
- ▶ Hand in the document before you leave today if possible. Otherwise due start of class on Day 2.
- ▶ Get help as needed from me

Handy for Growable Arrays HW

Properties of logarithms

$$\log_b(xy) = \log_b(x) + \log_b(y)$$

$$\log_b(x/y) = \log_b(x) - \log_b(y)$$

$$\log_b(x^\alpha) = \alpha \log_b(x)$$

$$\log_b(x) = \frac{\log_a(x)}{\log_a(b)}$$

$$a^{\log_b(n)} = n^{\log_b(a)}$$

Properties of exponents

$$a^{(b+c)} = a^b a^c$$

$$a^{bc} = (a^b)^c$$

$$a^b / a^c = a^{(b-c)}$$

$$b = a^{\log_a(b)}$$

$$b^c = a^{c \cdot \log_a(b)}$$