Announcements

- Hope you have been making progress on the homework
  - Hopefully you have already completed most of it

- Wednesday Feb 21 – Midterm
  - Open book
  - Will have a review session before hand

- Reading:
  - Chapter 3.1 – 3.4
Outline

- Feedback
  - Review of some things from last class
- Basic search
- Stack
  - code
- Queues
  - code
Java stuff

- Say you have a double/Double
- `double d = 3.123433445232399999`

- How do you print it out to only say 3.12 ??
String

- Naïve solution:
  - Turn into an array
  - Find decimal point (if any)
  - Process x numbers afterwards

- String class has a format method which takes specific formatting instructions

- Great if you know c
DecimalFormat

- Special class for formatting decimal numbers
- Bunch of options
- Let me show you an example
double d = 3.42422224;
System.out.println("this is d " + d);
DecimalFormat dc = new DecimalFormat();
System.out.println("Now its: 	"
        + dc.format(d));
dc = new DecimalFormat("###.##");
System.out.println("Now its: 	" +
        dc.format(d));
dc.applyPattern("000.00");
System.out.println("Now its: 	" +
        dc.format(d));
Node code

- Why node class
- How the code fits in
- Putting all the pieces together
Iterators

- Want to be sure you understand the idea

- Let me give you a really weird example

- Then we can look back at the code from last time
Applications

- Lists are everywhere
- Within programs
- On their own
- Part of larger algorithm

- Sorting is usually on a collection of items
  - Can anyone name a sorting routine??
Caution

- When sorting doesn’t all have to be same item class
- But should be comparable to each other
- Accomplish this by defining comparable behavior
Selection Sort

- Given an array of size n

- for every position 1..n
  - current position is the min spot
  - run through rest of your stuff until n
  - swap if less
public void selectionsort(int numbers[], int size) {

    int i, j, min, tmp;
    for (i = 0; i < size; i++) {
        min = numbers[i];
        for (j = i; j < size; j++) {
            if (min > numbers[j]) {
                tmp = numbers[j];
                numbers[j] = min;
                min = tmp;
            }
        }
    }
    numbers[i] = min;
}
run time?

- First time n
- Next time n-1
- Next time n-2
- ...
- Last time 1
We want to know

\[ \sum_{i=0}^{n} i = ?? \]
We want to know

\[ \sum_{i=0}^{n} i = \frac{n(n + 1)}{2} \]
Other kind of lists

- Sometimes would like to restrict the list in some way to be able to introduce some type of behavior

- For example, think of a pile of dishes
  - New dishes on top
  - When want to clean remove top dish
Question

- Lets say you want to write a math problem solver

- But want to make sure the equation is correct before starting

- Many types of parenthesis brackets
  - ()
  - []
  - {}
Goal

- Want to be able to say

- GOOD
  - \{ ( ) \}
  - \{ [ ]( ) ( ) \}

- Bad
  - ( { ) }
  - ( ( ) ]
Stack ADT

- 2 basic operations:
  - Push
  - Pop

- LIFO

- Think of it as being able to view only the top item in the list
Might think its pretty simple

Second most fundamental DS in computer science after arrays

Examples:
- When throw an exception we want to look at programming stack
- Think of running program as stack, when empty we are done
- Any idea on:
  - runtime ??
  - implementation ?
implementation

- Let me show you graphically

- array
  - how this works

- linked list
  - how this works
Supported functions

- Push
- Pop
- isEmpty()
coding

- What would the code look like for an array??

- Let me show you
  - Basic
  - ADT
Application

- balancing symbols

- let's write some code to see if we have balanced symbols in a block of text
Methods for Parenthesis Problem

- isOpenParent()
- isClosedParent()
- isParent()

- rule:
- if(isParent()){
  - if(isOpenParent())
    - push
  - else
    - pop and compare
    - now what ??
Anyone know what reverse polish notation is??
Postfix Notation

- postfix
- reverse polish notation

- infix
- prefix
Example 2

- Homework Example
  - can you write a palindrome detector

- Too bad, I hid a boot
mentioned it during recursive programming

- A calls B calls C

- LIFO
Queue DS

- List with specific property:
  - FIFO

- enqueue
- dequeue

- run time ??