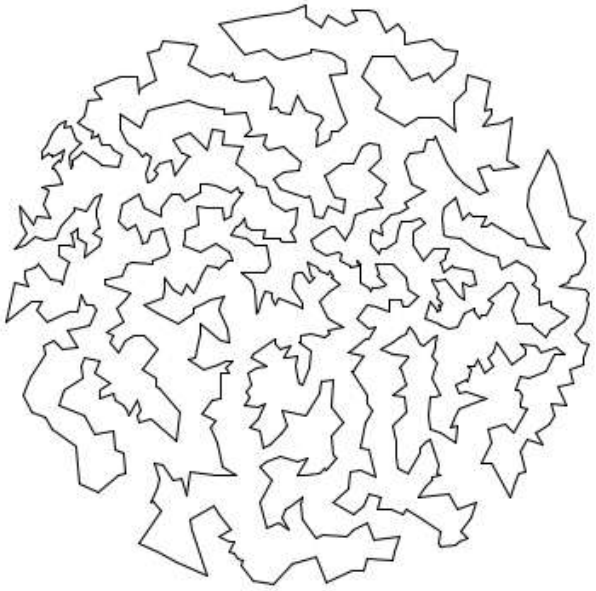
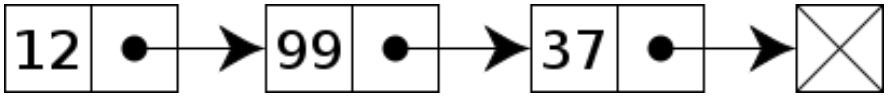
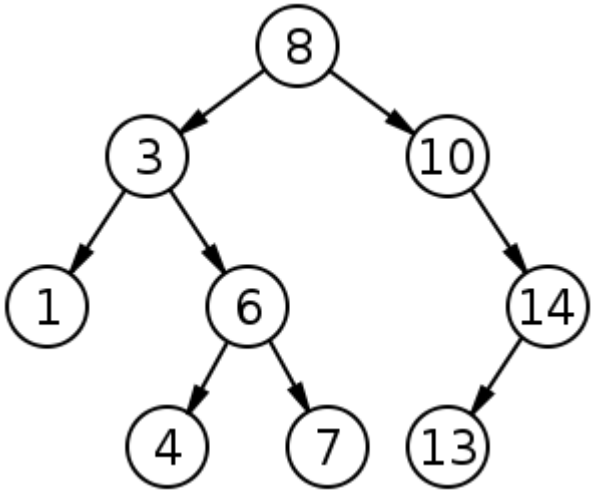
CSCI 136:
Fundamentals of Computer Science II







Keith Vertanen
Museum 102
496-4385
kvertanen@mtech.edu
http://katie.mtech.edu/classes/csci136

MontanaTech
THE UNIVERSITY OF MONTANA

Overview

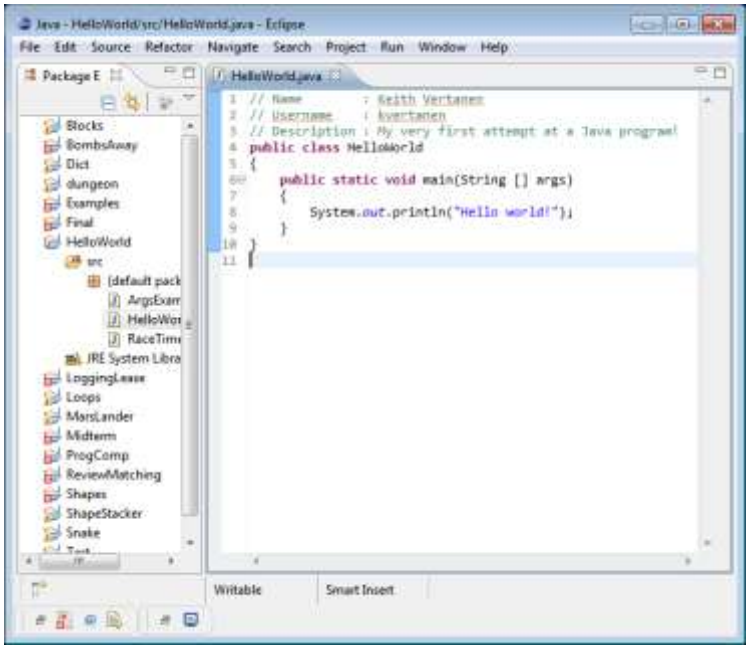
- Course details
  - People
  - Resources
  - Assessment
- Course topics

People

Keith Vertanen
Museum 102
496-4385
kvertanen@mtech.edu
Office hours: Mon 3:00-4:00p
Wed 2:00-3:00p
Fri 3:30-4:30p

Tyler Lee
Usually milling around Museum lab
trlee@mtech.edu
Lab TA hours: TBD

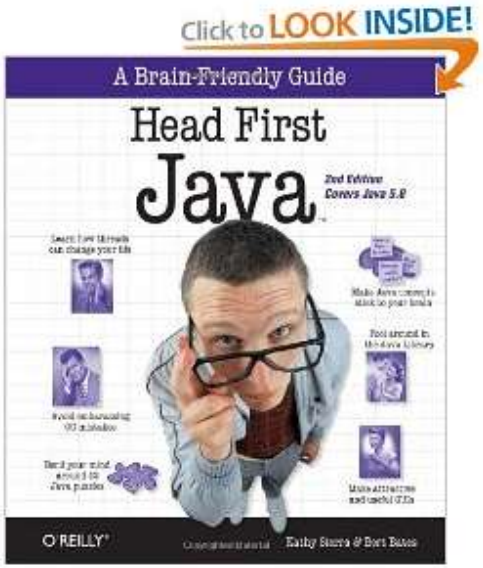
Development tools



- Eclipse IDE
  - On lab machines
  - Java 1.6
    - 1.7 not supported
  - Try and avoid standard input
    - Makes Tyler grumpy
    - We should learn to use the debugger
- Command line
  - DOS (Windows)
  - Unix (katie)
    - http://cs.mtech.edu/main/index.php/current-students/server-registration

Resources

- Textbook
  - We'll finish Head First Java
  - Chapters 12-18



- Laptop buddy program
  - I'm going to try integrating computer activities during (most) lectures
  - Work in pairs
  - Hopefully wireless will work…

Assessment

A	90% - 100%	Exam 1	15%	
B	80% - 89%	Exam 2	15%	
C	70% - 59%	Exam 3	15%	
D	60% - 69%	Programming assignments	55%	
F	0% - 59%	Staff discretion (participation and extra-credit)		±?%

- Assignments
  - Due 10PM on stated day, 2 hour grace period (<= 11:59pm)
  - 4 free late days, 0 if late and no free days left
  - Late days tied to student even if pair programming
  - Individual 1st half, optional pair programming 2nd half
  - Now out of 30 points
- Exams during scheduled 3-hour lab period
  - Written portion and/or computer portion

Where we've been

- We covered most of the Java language & various class APIs…
  - primitive data types
  - boolean expressions
  - if-else statements
  - switch-case
  - for-loop
  - enhanced for-loop
  - while-loop
  - do-while loop
  - arrays, 1D, 2D
  - static methods
  - instance variables
  - instance methods
  - try-catch
  - enumerations
  - Math
  - String
  - ArrayList
  - Double
  - Integer
  - PrintWriter
  - File
  - Scanner

Course topics

- 1: Data structures
  - How we store and organize data in our programs

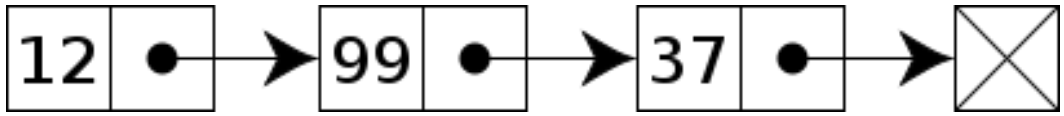
> "I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships."
>
> -Linus Torvalds, creator of Linux

  - Opens up the types of programs we can build
    - Not everything works if shoved in an array!

Data structures

- Set
- Queue
- Stack
- Linked list
- Tree
- Graph
- Hash table

```
private class Node
{
    private int num;
    private Node next;
}
```



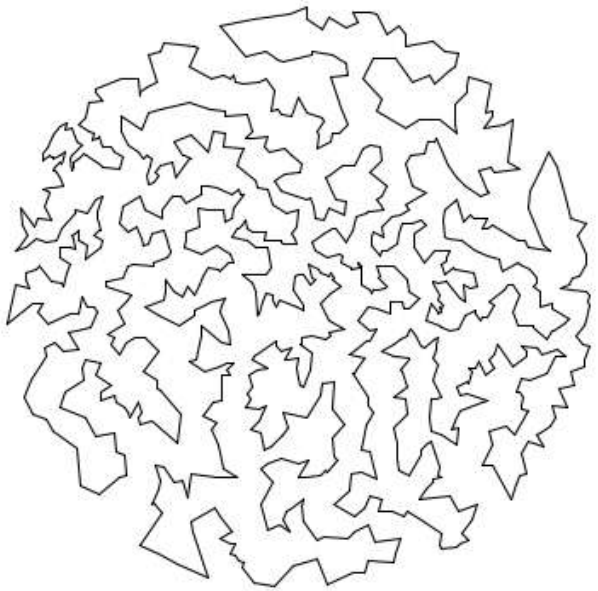
Course topics

- 2: Algorithms
  - Steps we take to solve a problem
  - Embodies the cleverness to solve problem correctly and efficiently
  - Smart algorithm + right data structure
    - Makes the seemingly impossible possible
    - (but still can't do everything)

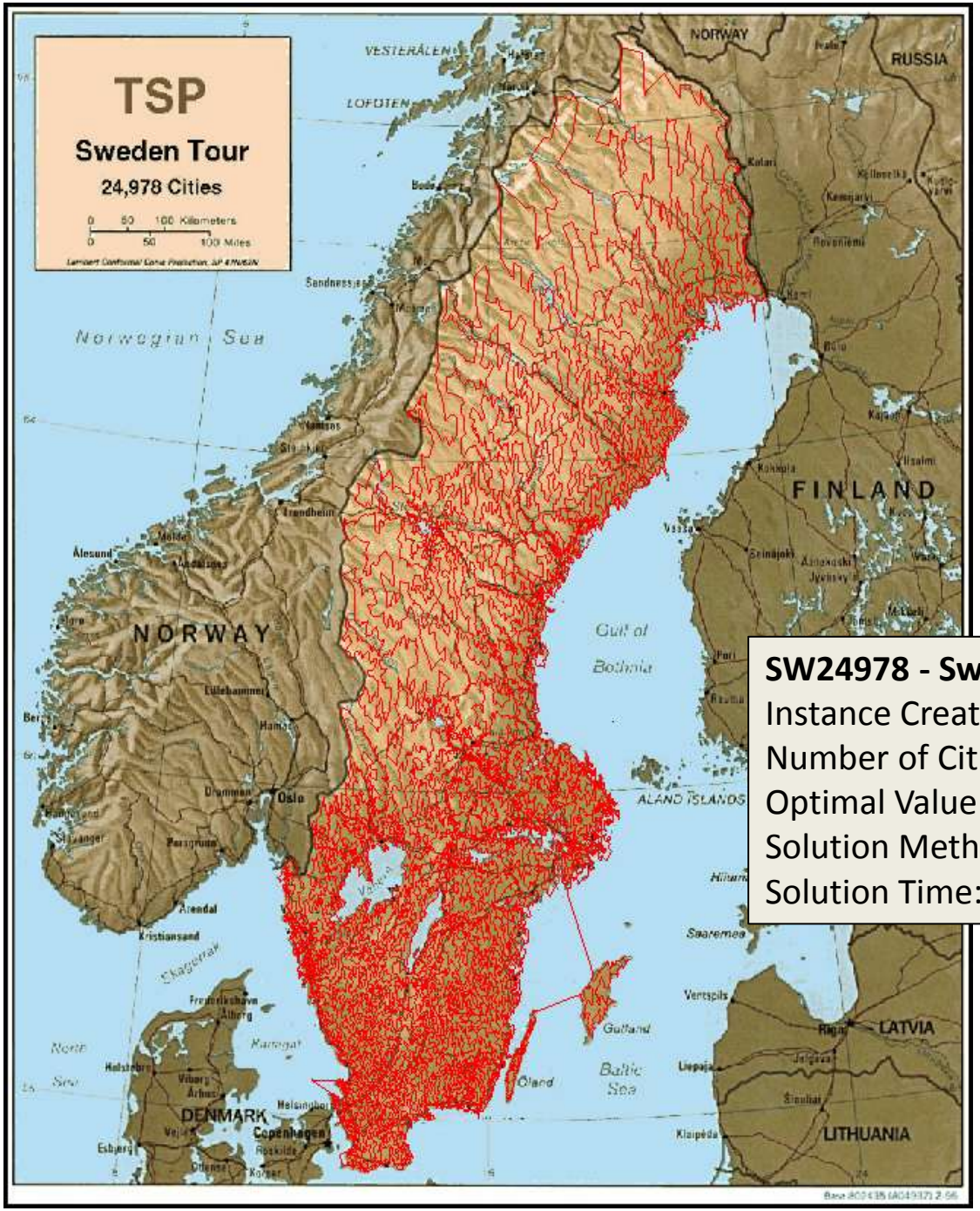
Travelling salesman

- Travelling salesman problem (TSP)
  - Locations of a bunch of cities
  - Find shortest possible tour visiting each city exactly once

1000 cities



optimal tour



SW24978 - Sweden Computation Log

Instance Created: July 29, 2001

Number of Cities: 24,978

Optimal Value: 855,597

Solution Method: Concorde, CPlex 6.5 LP Solver, LKH

Solution Time: 84.8 years, Intel Xeon 2.8 GHz

http://www.tsp.gatech.edu/sweden/index.html

Travelling salesman

- Travelling salesman problem (TSP)
  - Finding optimal tour is easy
    - Try all possible combinations
      - Exponential in number of cities
    - This takes an enormous amount of time!
  - Can we find optimal tour faster?
    - Most people think the answer is no
    - No one has proved it

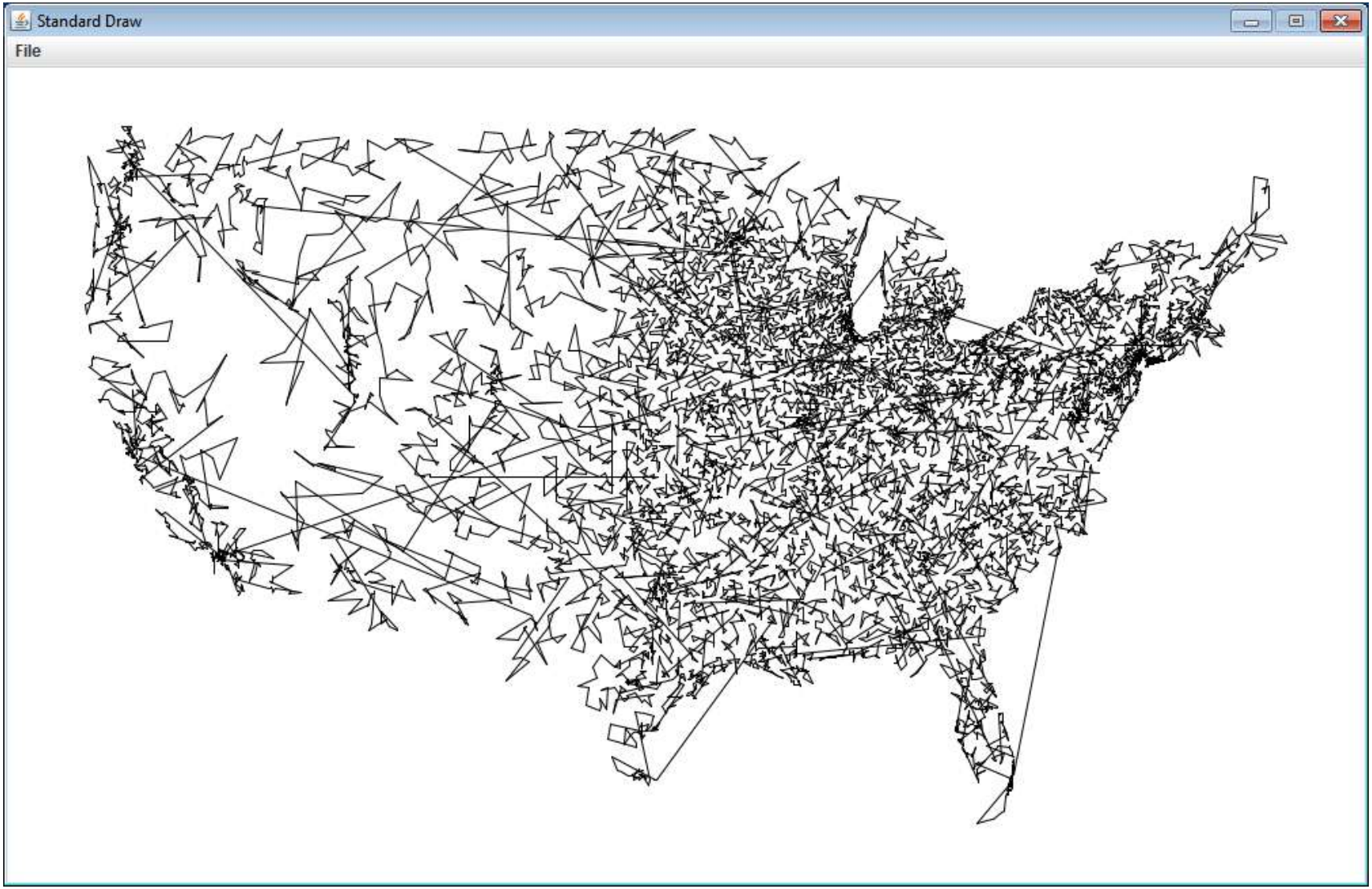
TSP algorithm 1

- Approximate solution
  - Data structure = linked list
    - Makes it quick to insert next city anywhere in the list
  - Algorithm = add city into list next to closest existing city
    - Heuristic, not provably optimal but usually does okay

```
536.6211   476.8667
716.6871   433.0017
505.1939   323.8175
613.9327   443.7259
694.1236   218.8665
819.1546   396.5130
...
```

File with locations of 13509 US cities.

Algorithm 1: nearest neighbor



Tour distance = 77449.98

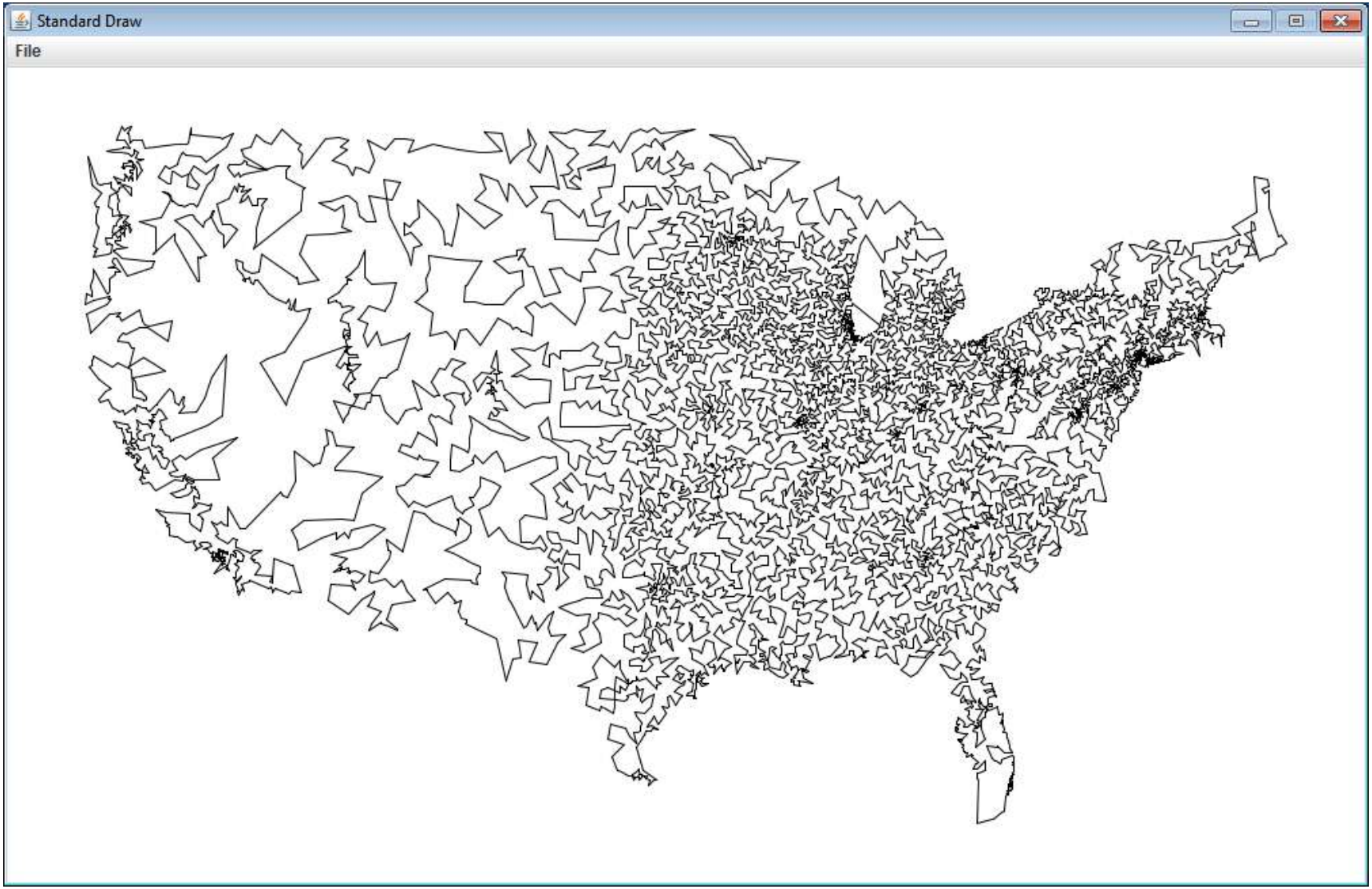
TSP algorithm 2

- Approximate solution
  - Data structure = linked list
    - Makes it quick to insert next city anywhere in the list
  - Algorithm = add city into list wherever it causes least increase in total tour length
    - Heuristic, not provably optimal but usually does okay

```
536.6211   476.8667
716.6871   433.0017
505.1939   323.8175
613.9327   443.7259
694.1236   218.8665
819.1546   396.5130
...
```

File with locations of 13509 US cities.

Algorithm 2: smallest increase



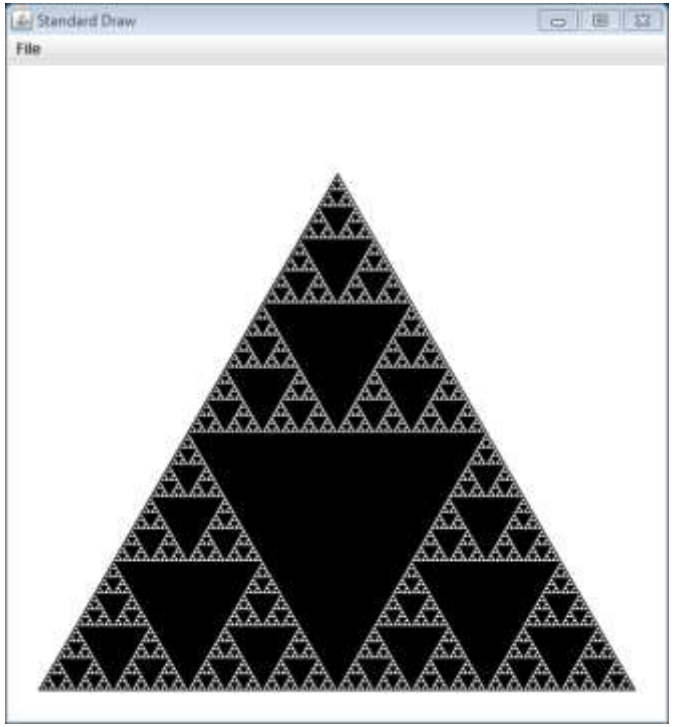
Tour distance = 45075.78

Course topics

- 3: Recursion
  - Methods calling themselves
  - Often useful technique for solving a problem
    - Non-recursive solution is always possible, but code would be harder and longer

```
public class Factorial
{
   public static long fact(long n)
   {
      if (n <= 1)
        return 1;
      return n * fact(n - 1);
   }

   public static void main(String [] args)
   {
      System.out.println("4! = " + fact(4));
   }
}
```



Training a language model

Call me Ishmael. Some years ago- never mind how long precisely- having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet; and

What letter comes next???

It was a_

I want to go to t_

n-gram	count
aa	4
…	
am	1695
an	10435
ao	14
…	
tg	2
th	15570
ti	4246
…	
zz	42

Need to map an n-gram String to an integer count.

data structure = binary search tree

Using the language model

- Message corrupted in transmission
  - Given language model
  - Recursive algorithm to find most likely messages

```
% java FixCorruptedNbest 5 5 "it wa~~t~~~b~~t~o~~t~m~s~
~t~~as~t~~~~ors~~~~ ti~~~." < wiki_100k.txt

Noisy   : it wa~~t~~~b~~t~o~~t~m~s~ ~t~~as~t~~~~ors~~~~ ti~~~.
Nbest 0 : it was the best of times, it was the worst of times.
Nbest 1 : it was the best of times, it was the worships title.
Nbest 2 : it was the best of times, it was the worsenic times.
Nbest 3 : it was the best of times, it was the horse of times.
Nbest 4 : it was the best of times, it was the worsenic title.
```

Course topics

- 4: Threads
  - One program with multiple threads of execution
    - e.g. background thread to animate progress bar while main program downloads file
  - Sometimes can help simplify program
    - Spawn thread to do a job, go back to something else
  - Processors no longer getting faster
    - Instead they add more and more cores, 2, 4, 6, …
    - A multi-threaded programs may be able to use multiple cores at same time, getting job done faster

Course topics

- 5: Socket communication
  - Send information between two programs
    - On the same computer
    - On computers next to each other
    - On computers on different sides of the globe
  - e.g. Building a multi-player network game

Course topics

- 6: Graphical User Interfaces (GUIs)
  - Building interfaces with buttons, etc.
  - Dealing with events
  - Learn to do draw ourselves rather than relying on StdDraw

Audience participation

- What topics are missing?
  - Publish programs (deployment)
  - Cross-platform
  - Mobile app?