

DEPTH + BREADTH

- **Goals:**
 - Provide science students with necessary computational skills.
 - Make CS exciting for science-oriented students (and lure some into CS).
 - Prepare potential CS majors.

- **Depth:** problem solving and programming in Python
 - applications to scientific problems

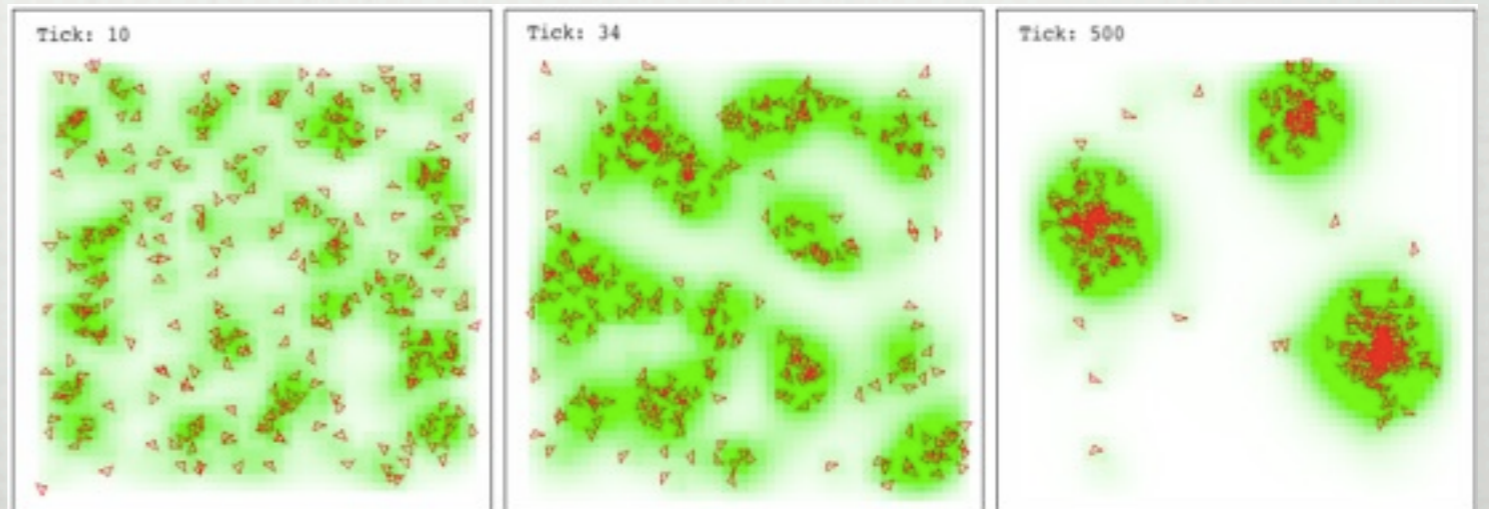
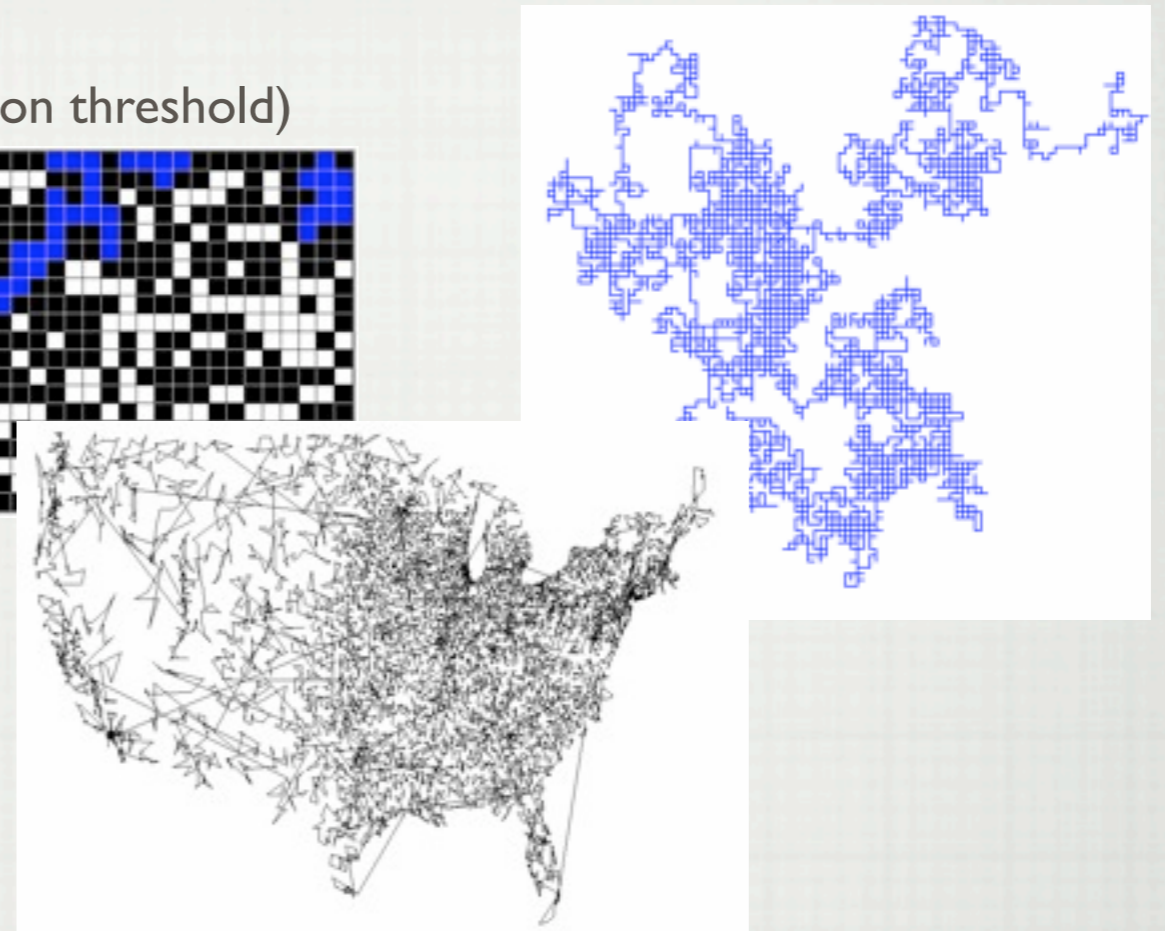
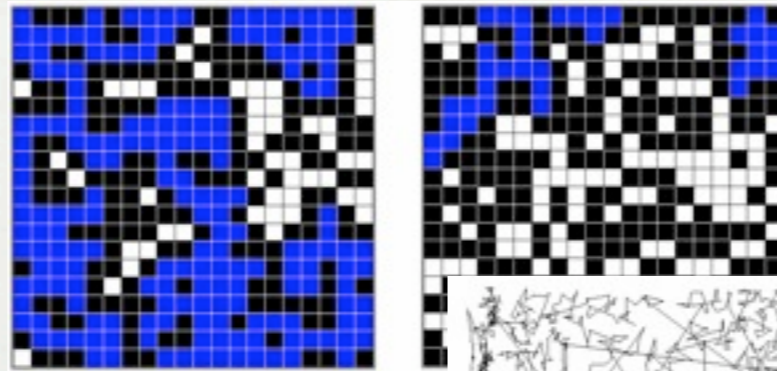
- **Breadth:** Big Idea Tuesdays
 - discussions of books and articles
 - ties into programming projects

THE USUAL SUSPECTS

- Algorithms, abstraction, modularity
- Problem solving: patterns, simplification, generalization
- Data types, arithmetic, assignment
- Using objects, methods, and modules
- Iteration: for and while loops, accumulators
- Writing functions, parameters, return values
- Boolean expressions and conditional constructs
- String, List, and Dictionary objects and methods
- Simple sorting algorithms
- Introduction to time complexity
- Working with files and data on the web
- Recursion
- Writing classes and object-oriented programming

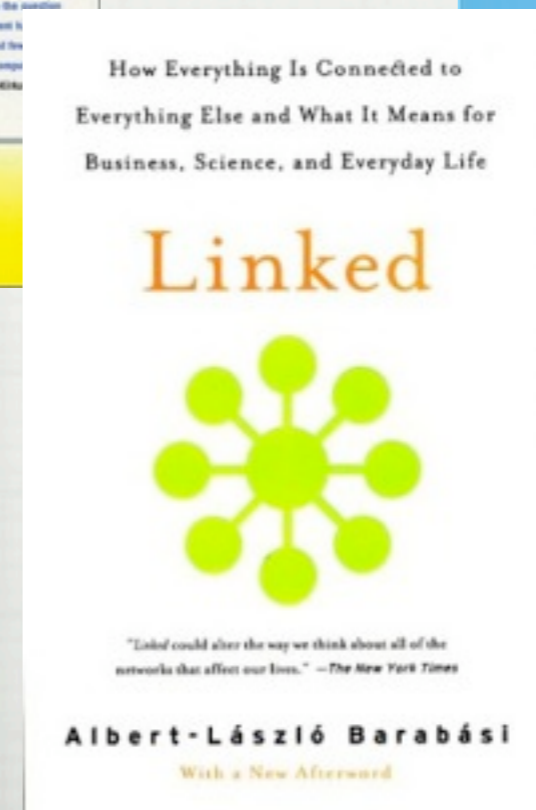
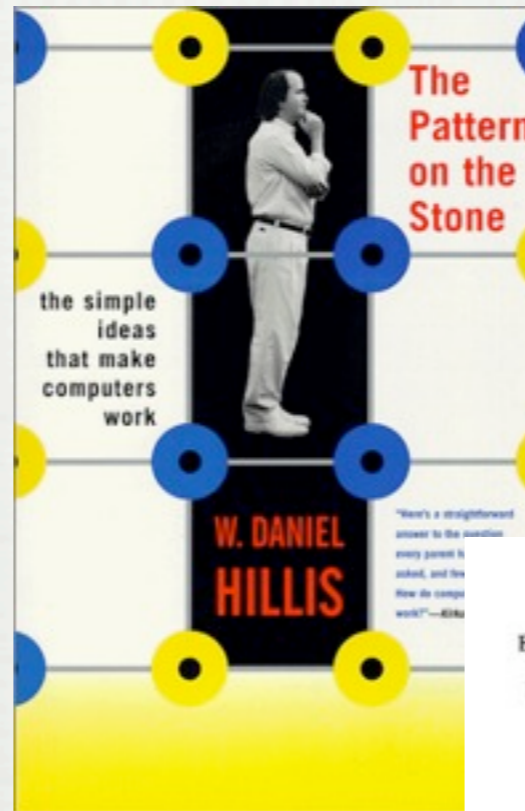
APPLICATIONS AND PROJECTS

- Monte Carlo simulations (random walks, percolation threshold)
- Histograms and list statistics
- DNA algorithms
- Genetic algorithms
- k-means clustering
- Networks (graphs): searches and shortest paths, small world phenomena, scale-free properties
- Traveling salesperson heuristics
- Fractals and Lindenmayer systems
- Emergence and swarm intelligence
- n -body simulation



BIG IDEA TUESDAYS

1. What is computation?
2. Limits of computation
3. Genetic algorithms
4. Networks
5. Quantum computing
6. Digital forensics
7. Swarm intelligence



OTHER POSSIBLE BOOKS

