Programming projects for Intro Scientific Programming

Victor Eijkhout
eijkhout@tacc.utexas.edu
https://tinyurl.com/vle322course

CppCon 2021

1. In a nutshell

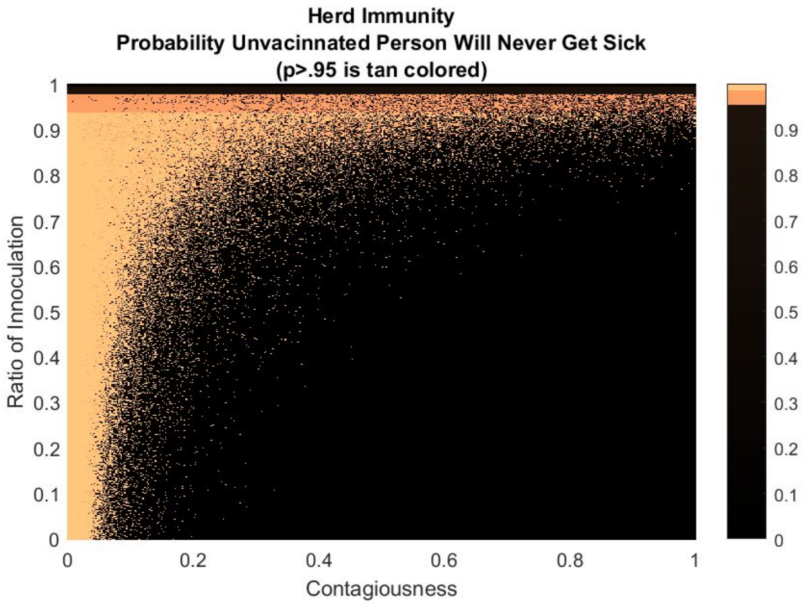
- I teach C++ (and Fortran2008) to engineering-type students the The University of Texas at Austin
- End-of-semester programming project instead of exam
- Do a semi-realistic scientific simulation
  Write up your findings as if research article

2. Why this presentation?

- Share and enjoy! `https://tinyurl.com/vle322course`
- Invite contributors and collaborators.
  `mailto:eijkhout@tacc.utexas.edu`

3. Project: infectious disease simulation

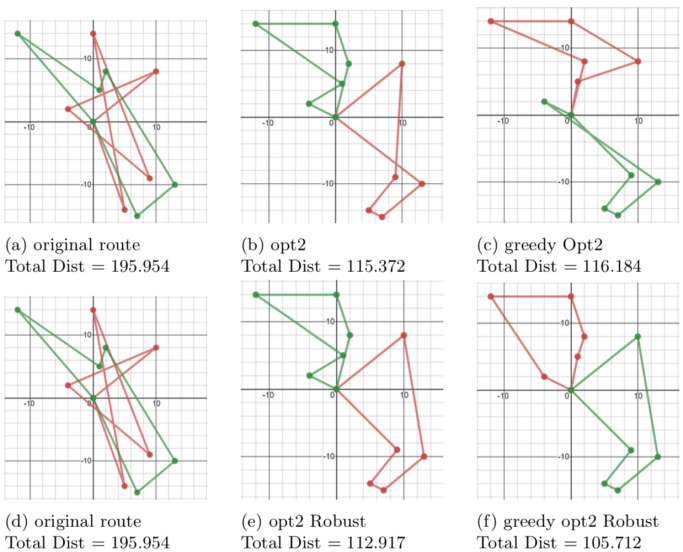
- How does an infectious disease spread through the population? Does anyone escape being infected? How long does the disease run?
- Investigate influence of parameters: chance of transmission, incubation period, how many people are vaccinated, ...
- Programming: basic OO, great for C++ 101



- Sample graph:

4. Project: Amazon delivery trucks

- How do you plan an optimal route for a delivery truck?
  How about if you have more than one truck?
- Investigate heuristics for route planning.
  Discuss management and ethics issues.
- Programming: Multiple Traveling Salesman Problem
- Sample graph:



(a) original route
Total Dist = 195.954

(b) opt2
Total Dist = 115.372

(c) greedy Opt2
Total Dist = 116.184

(d) original route
Total Dist = 195.954

(e) opt2 Robust
Total Dist = 112.917

(f) greedy opt2 Robust
Total Dist = 105.712

5. Project: redistricting

- Redistricting / Gerrymandering
- Group census districts into congressional districts.
- Is it possible for a minority to gain the upper hand? Is it possible to prevent this?
- Programming: dynamic programming

6. Project: Google Pagerank

- Simulate the internet
  Which web pages are important?
  Search-Engine Optimization
- Different techniques for modeling the problem.
- Dig into the mathematics of it: relation between graphs and sparse matrices.
- Programming: DAG vs Sparse Matrix

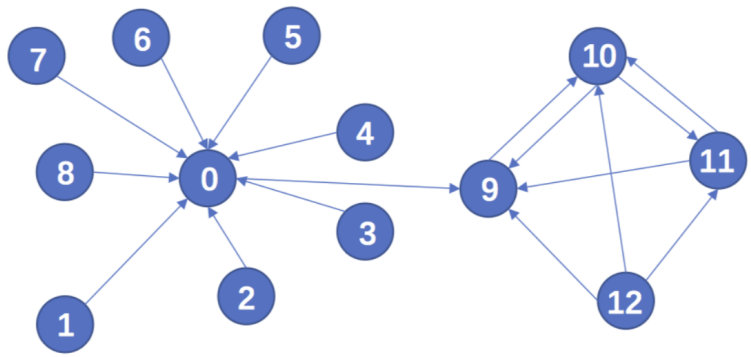


Figure 6: Web with one page artificially made 'important'

- Sample graph:

7. Project: High performance linear algebra

- Linear algebra algorithms that are optimized for architecture
- Cache-oblivious strategy: `std::span`