Scientific Computing

Announcements

→ Office Hours: Mondays + Fridays, 9:30 - 10:30 Cudahy 307

→ HW 1 assigned
On D2L → Dropbox
Due Friday, Jan 31

* Acceptable Sources: Online searches for how to do things in Python cite!

Unacceptable: Searching for the questions, AI Tools

Today

→ Coding Exercises

→ Greedy Algorithms

Topic 1b - Jupyter vs .py files

* Jupyter is good for playing around and for presenting, but not for final products.

* It lets you run code out of order.

[demo]

* I strongly recommend <u>against</u> doing your work in Jupyter and then copy+pasting into a .py file. Always leads to problems.

* You can write your code in VS Code and run it in a terminal right inside VS Code. This is a "unix" terminal, same as a Mac, Linux, and the "Git for Windows" terminal.

↳ Makes this not necessary anymore.

demo: Find the smallest positive integer that when you square it, the digits start with 2024

Topic 2 - Greedy Algorithms

A lot of topics we'll cover in this class fall into the category of problem solving paradigms — a catalogue of ways to approach new problems.

"heuristics"

What is a problem? — super vague
You might have input data and/or constraints.
The question might be:
- Is it possible to satisfy all the constraints?

Ex: Every year, the NFL has to come up with a season schedule. There are many constraints! 17 games

- 32 teams in 2 conferences of 16 teams each
- Each conference split into 4 divisions of 4 teams each

Each team plays:

- 3 division rivals, twice each (H+A)
- each team in another division in the same conference (2H+2A)
- five teams in the other conference
- two more teams in their own conf.

plus:
- stadium constraints
- TV constraints
- holiday games
- one week off between Week 6 and 14
- many more...

Question: Can this be done?

(Obviously, yes)

The NFL says it takes 1000s of computers to generate ~1000 valid schedules, and then humans pick the best one.

Instead of "Is it possible?" the question could be "What's the optimal solution?"
(lowest cost, highest profit, etc)

Ex: If Amazon has 100 packages to deliver to different houses in Milwaukee, and 5 delivery vans, which route
- uses the least gas
- travels the fewest miles
- takes the least time

etc.

Greedy Algorithms

Vague definition: A greedy algorithm is a way of solving a problem that builds up a solution bit by bit, always picking the next bit that is the best, even that leads to a suboptimal full solution.

"neurotic"

They are:
- normally lightning fast
- much better than random solutions
- sometimes pretty bad, sometimes pretty good, sometimes provably optimal, depending on the problem.

Ex: Giving change - How does a cashier give change? Suppose you owe $3.27 and pay with $20. They start giving you bills and coins from largest to smallest.

$20 - $3.27 = $16.73

~~$100~~ ~~$50~~ ~~$20~~ $10 5 1 0.25 0.10 ~~0.05~~ 0.01

1 1 1 2 2 3

$6.73 $1.73 $0.23 $0.03 $0.

$0.73

$10, $5, $1, 2Qs, 2Ds, 3Ps

= 10 items