Programming, Problem Solving, and Algorithms

CPSC203, 2019 W1

Announcements

Project 3 released soon. Due 11:59p, Nov 29.

“Problem of the Day” continues!

Today:

Sudoku Implementation - one last thought

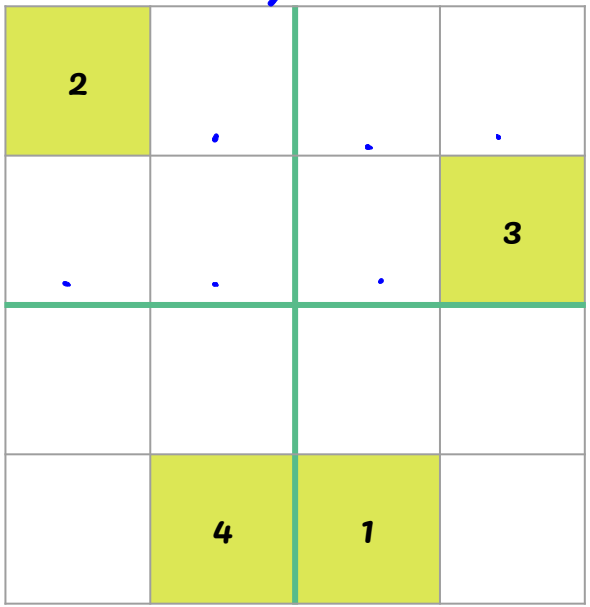
Maps!

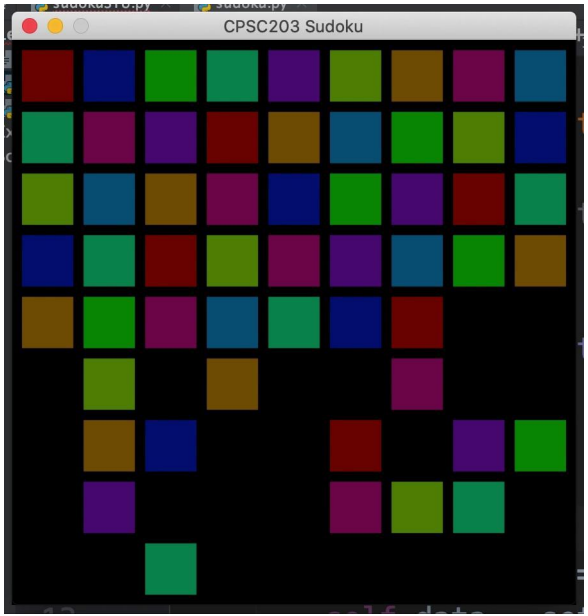
Shortest Path

Sudoku, one last thought...

Recall our algorithm for searching... could we be smarter?

DFS



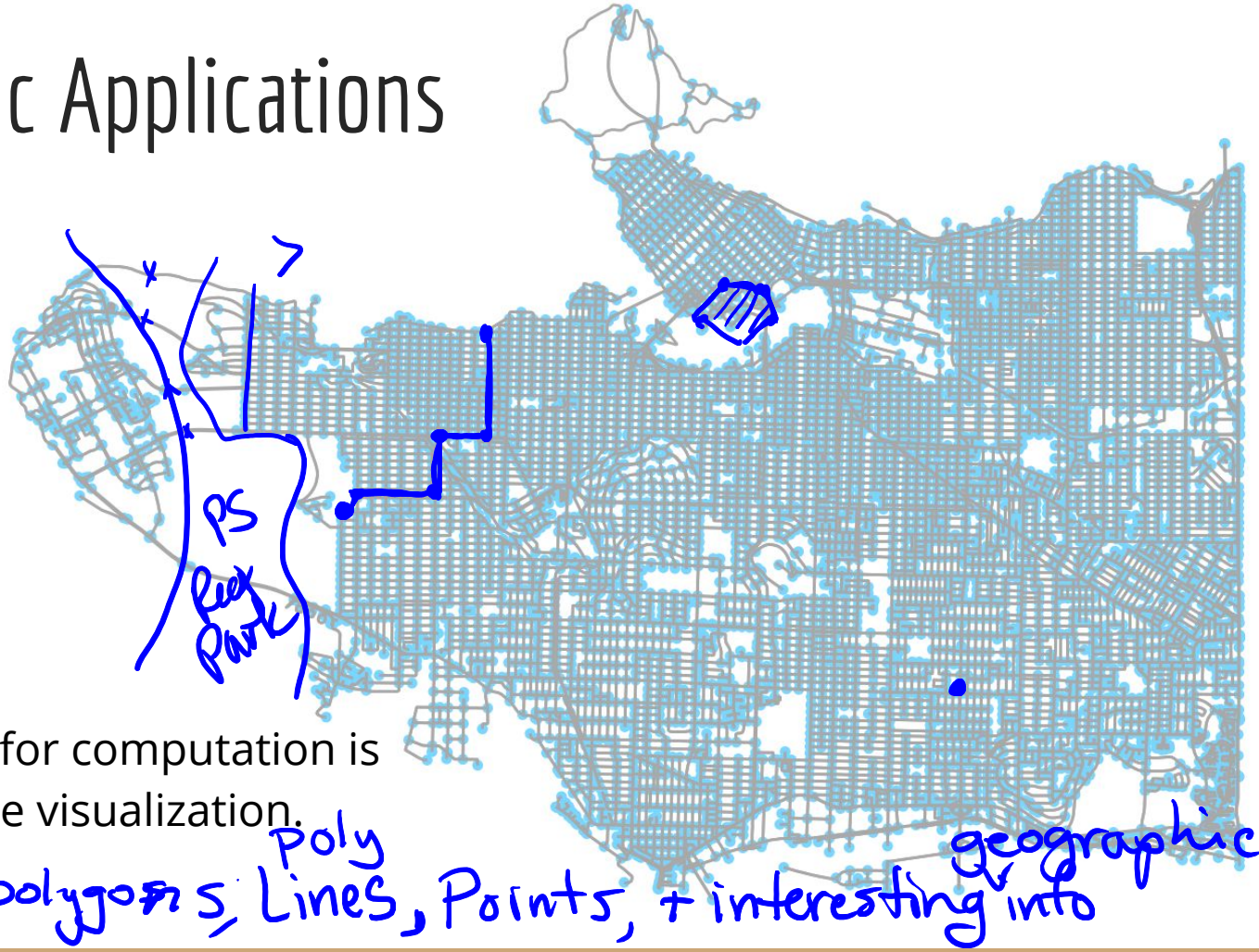


algorithmic.

apply rich improv
based on iterative choices

choose a promising direction

Geographic Applications



The data we use for computation is separate from the visualization.

Data: graph, polygons, Poly Lines, Points, + interesting info geographic

Open Street Maps

An open-source alternative to Google Maps' *data*.

https://www.openstreetmap.org/directions?engine=fossgis_osrm_car&route=49.2643%2C-123.1772%3B49.2584%2C-123.2466#map=14/49.2593/-123.2119

OSM provides an Application Programmer's Interface (API) that allows our program to request data, which is returned in a reasonable format.

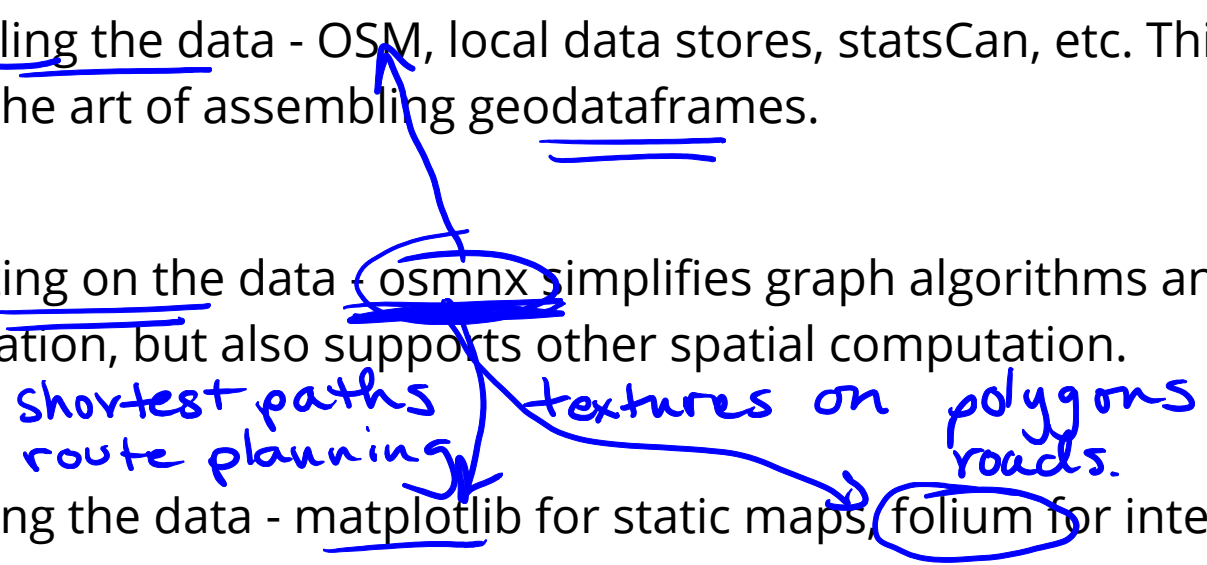
Example: `ox.gdf_from_places(place_names,gdf_name='UBCVan')`

	geometry	place_name	bbox_north	bbox_south	bbox_east	bbox_west
0	POLYGON ((-123.26221 49.26737, -123.26178 49.2...	University of British Columbia, West 16th Aven...	49.273124	49.243131	-123.227362	-123.262213
1	POLYGON ((-123.24492 49.27961, -123.24467 49.2...	Pacific Spirit Regional Park, West 16th Avenue...	49.279788	49.235248	-123.193671	-123.244925
2	POLYGON ((-123.22496 49.27462, -123.22475 49.2...	Vancouver, Metro Vancouver Regional District, ...	49.316171	49.198445	-123.023242	-123.224961

Map applications

Three parts:

1. Assembling the data - OSM, local data stores, statsCan, etc. This is mostly the art of assembling geodataframes.
2. Computing on the data - osmnx simplifies graph algorithms and computation, but also supports other spatial computation.
3. Visualizing the data - matplotlib for static maps, folium for interactive maps.



Introductory Demo

What surprises you in the code?

_____

_____

What surprises you in the maps?

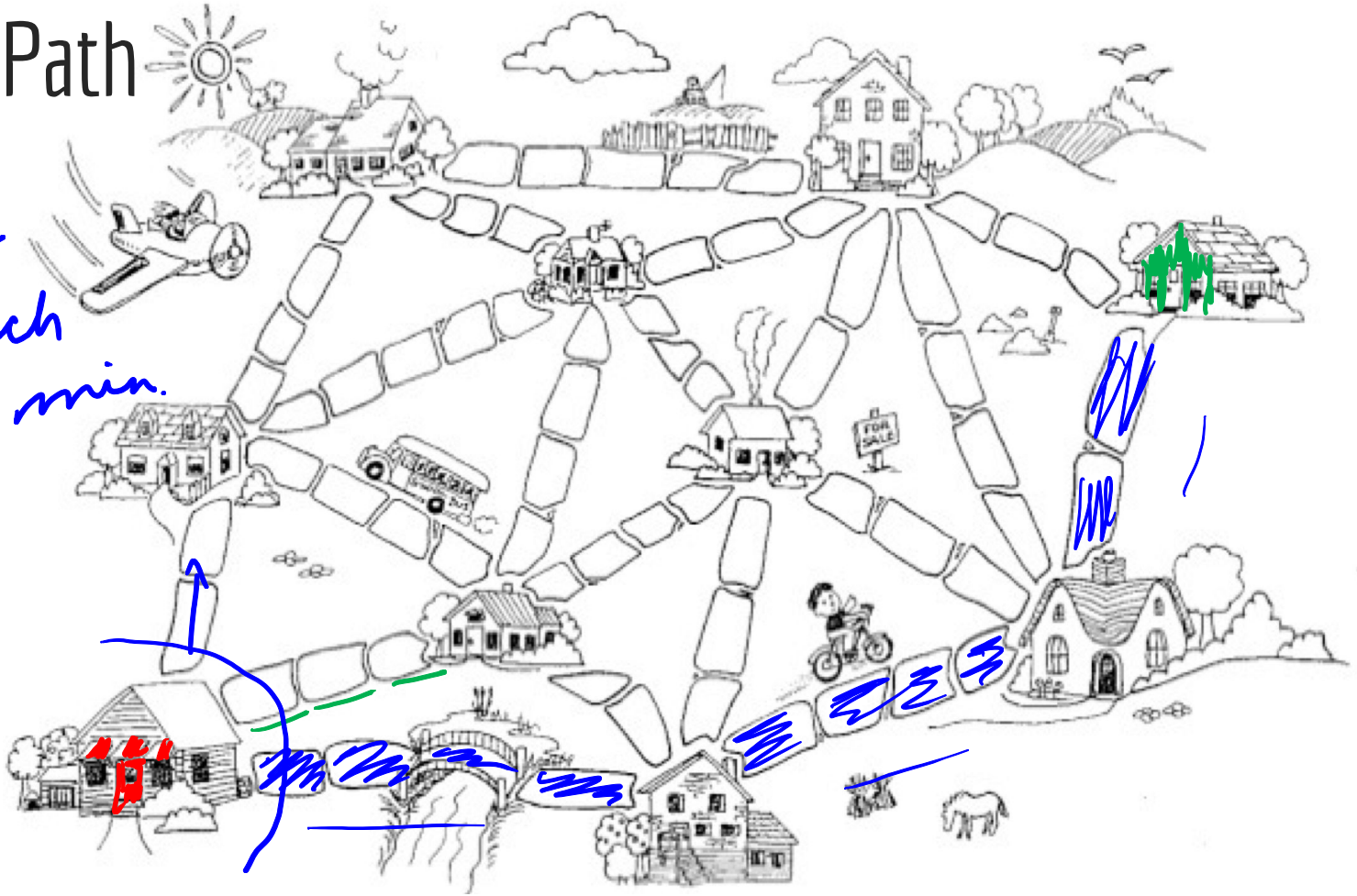
_____

_____

Shortest Path

How long to get from ■ to ■ if each ▭ takes 5 min.

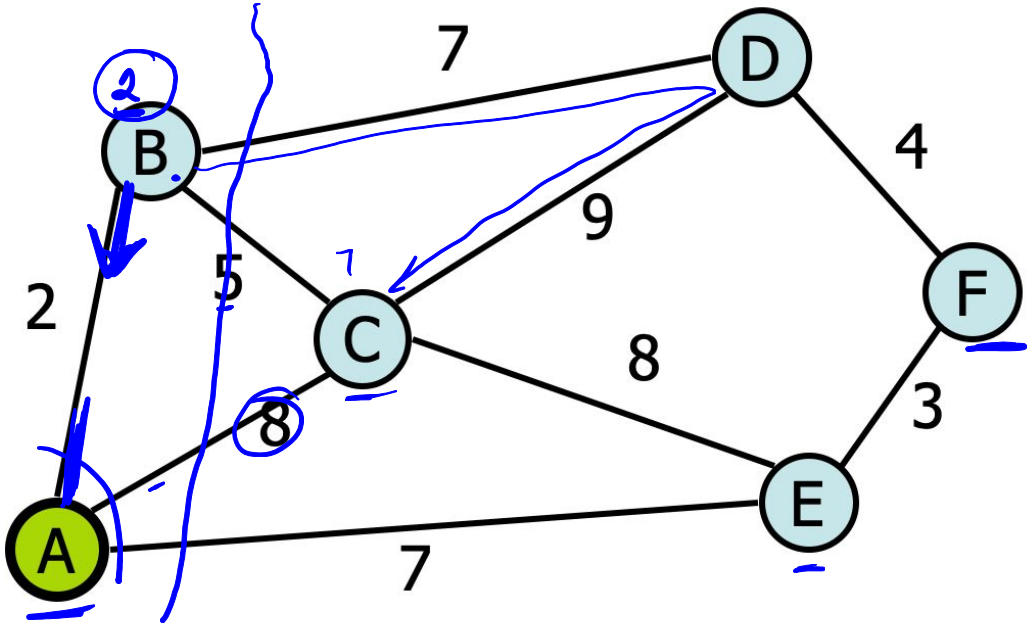
590 min



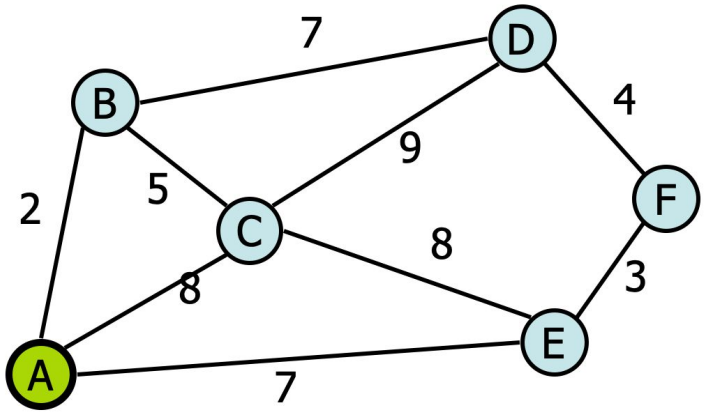
Dijkstra's Algorithm

Single Source Shortest Path: Given a graph G and a start vertex s, returns the shortest path from s to every other vertex in G.

Next time we'll clarify + finish.



Dijkstra's Algorithm



Initialize structure:

1. For all v, d[v] = "infinity", p[v] = null
2. Initialize source: d[s] = 0
3. Initialize priority (min) queue
4. Initialize set of labeled vertices to ∅.

Repeat these steps n times:

- Find & remove minimum d[] unlabelled vertex: v
- Label vertex v
- For all unlabelled neighbors w of v,

  If cost(v,w) < d[w]

  d[w] = cost(v,w)

  p[w] = v

Dijkstra's Algorithm

How is this algorithm similar to BFS/DFS?

_____

_____

How is this algorithm different than BFS/DFS?

_____

_____

Initialize structure:

1. For all v, d[v] = "infinity", p[v] = null
2. Initialize source: d[s] = 0
3. Initialize priority (min) queue
4. Initialize set of labeled vertices to ∅.

Repeat these steps n times:

- Find & remove minimum d[] unlabelled vertex: v
- Label vertex v
- For all unlabelled neighbors w of v,

  If cost(v,w) < d[w]

  d[w] = cost(v,w)

  p[w] = v

POTD #34 Tue

https://github.students.cs.ubc.ca/cpsc203-2019w-t1/potd34

Describe any snags you run into:

1. Line ___: _____
2. Line ___: _____
3. Line ___: _____
4. Line ___: _____
5. Line ___: _____

ToDo for next class...

POTD: Continue every weekday! Submit to repo.

Reading: TLACS Ch 10 & 12 (lists and dictionaries)

References:

https://www.youtube.com/watch?v=wsSEKm-rU6U

https://github.com/gboeing/osmnx-examples/tree/master/notebooks

https://gist.github.com/psychemedia/b49c49da365666ba9199d2e27d002d07