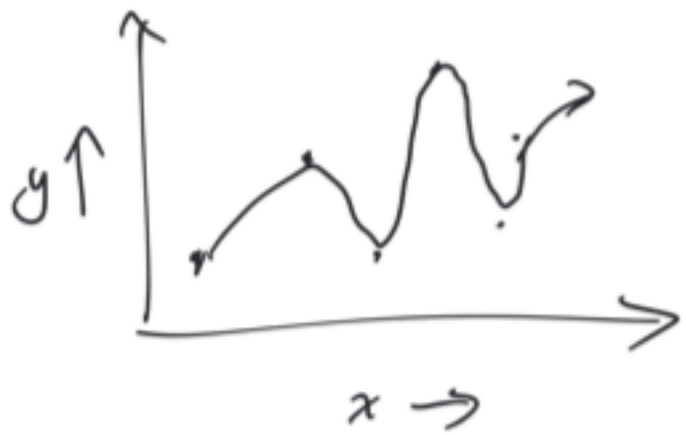
Lecture 19: Graphs

- Graph definitions
- Adjacency List / Adjacency matrix

Announcements

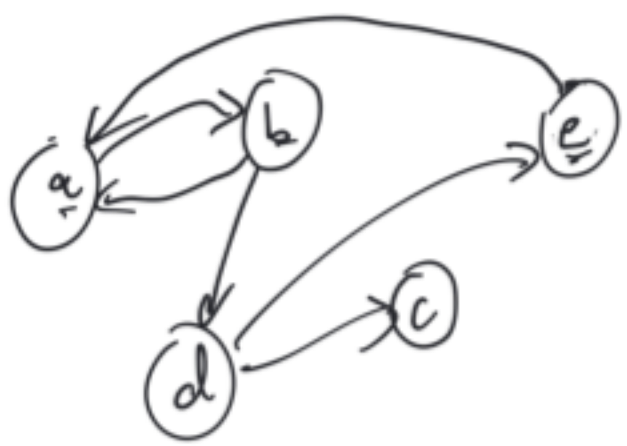
- Visiting lecture tomorrow after class
- assignment specs



not what we mean when we say "graph"

"directed" graph: collection of <u>vertices</u> (points) with <u>edges</u> connecting them

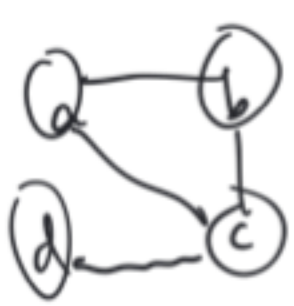
directed edge (from u to v)



vertices: $\{a, b, c, d, e\} = V$

edges: $\{(a,b), (b,a), (b,d), (d,c), (d,e), (e,a)\} = E$

"undirected": edges don't have a direction



Graph examples

- Map (subway system, transportation between warehouses)
  - vertices: stations
  - edges: connections

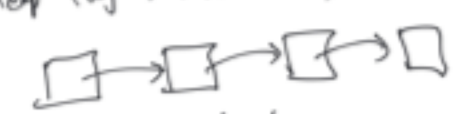
- physical structures in 3-D space
  - vertices: points in space
  - edges: beams connecting them.

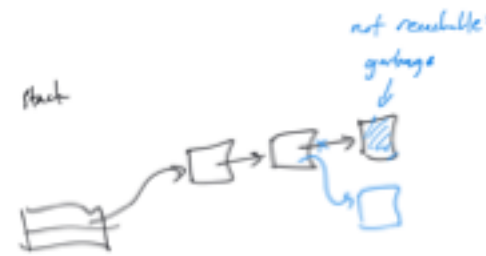
- Web sites:
  - vertices: web pages
  - edges: links from one page to another.

- Brain:
  - vertices: neurons
  - edges: connections between neurons.

- Social network:
  - vertices: people
  - edges: relationships between people (e.g. "friends")

- Heap (objects & references) as a graph





  - vertices: objects
  - edges: references from one to another.

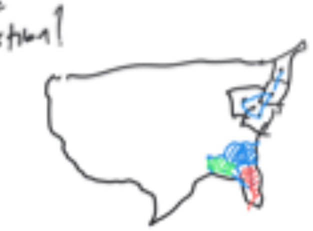
- Schedule exams:
  - vertices: classes
  - edges: edge between two classes if there is a student in both

Goal: assign an exam slot to each vertex, such that adjacent classes have different times

adjacent ← there is an edge from one to another.

Same question!

- Map coloring:
  - vertices: states
  - edge between two states if they share a border.



would like to color each state so that adjacent states don't have same color.

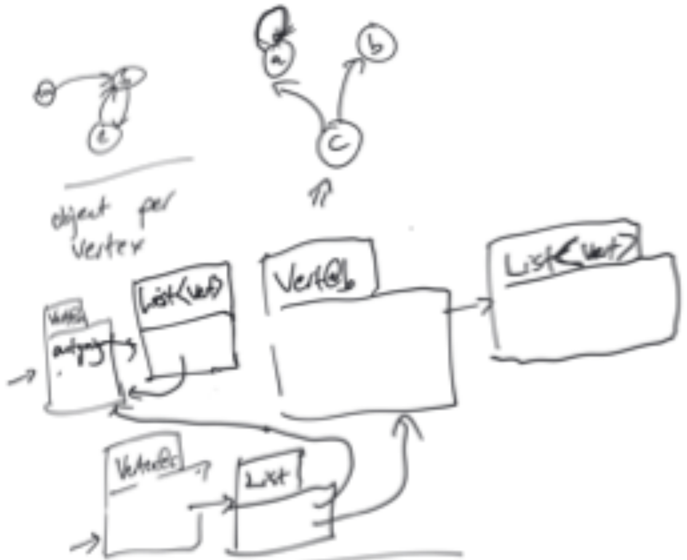
Graph as data structure:

- add / remove vertices / edges
- from any vertex: iterate over outgoing / incoming edges
- are vertices u, v adjacent?
- how many verts / edges exist?

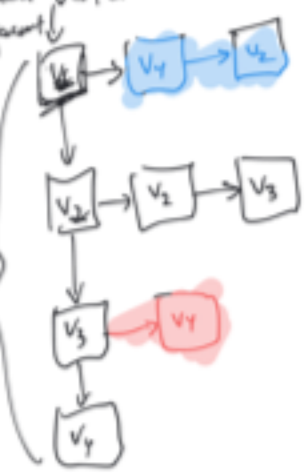
Labeled graph: there is some data associated with each edge

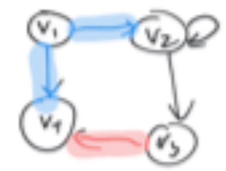
- Map: put travel time along each edge
- Social network: kind of relationship
- Neural network: strength of connection
- Communication: capacity

- associate data with a given edge.



adjacency list: each vertex has list of adjacent verts





array, not containing "v_1, v_2, ..."

any collection (array, HashSet)

linked list of vertices, each containing a list of neighbors,

Store all vertices in a set,

Store all edges in a set.

adjacency matrix: 2-D table, which has one row, one column per vertex

Question	How good with adj. list?	How good w/ adj. matrix?
Storage space?	$O(\lvert V\rvert + \lvert E\rvert)$ one obj. for each vert., each edge	$O(\lvert V\rvert^2)$ one entry for each pair of vertices

dense: $|E| \approx |V|^2$

Sparse: $|E| \ll |V|^2$

what's largest # of edges in a graph with $|V|$ vertices?

in worst case, every vertex has an edge to every other,

up to $|V|^2$ edges.

Complete graph

clique: everything adjacent to everything else

In many applications, vertices are adjacent to a constant # of other vertices.

Important operation: find all vertices u reachable from a given vertex v (maybe find a path from v to u.

(graph traversal, tomorrow)