Lecture 23: Graph algorithms

- Topological sort
- Coloring
- Planarity + 4 color theorem
- Quiz

Nodes: tasks   Edges: prerequisites

before starting T1, must complete T2

e.g. tasks are courses.

Topological sort: an ordering plan of vertices where if $i < j$ then there is no edge plan[i] → plan[j]

i.e. all prerequisites complete before starting task i.

To start: find a course with no prereqs
vertex " " out edges.

loop invariant:

all outgoing edges in plan only point to vertices already in plan

in-degree(v) = # of edges into v

out-degree(v) = # of edges out of v

degree(v) = # of edges connected to v.

top. sort(G)

1. find vertex v with no outgoing edges
   i.e. with out-degree 0.
2. add v to path.
3. remove v from G (and all edges to/from v)
4. topologically sort G.

Graph coloring

- Graph represents a map of $\frac{\text{countries}}{\text{vertices}}$
- Edge c_1 to c_2 if they share a border.

- Goal: assign a color to each country such that adjacent countries have different colors.
- Important goal: how many colors are necessary?

Colors:
0
1
2
3
4
⋮

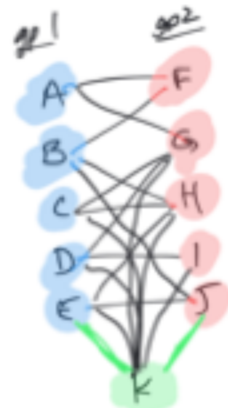
How to color a graph?
- pick any vertex, color it.
- pick any other vertex, pick any color different from colors of all neighbors.

(first unused color possible)
- smallest color c with none of v's neighbors having color c

invariant: colored portion is a valid coloring

Could have one color per vertex

complete graph



bipartite:
two groups of vertices, groups 1 & 2 with edges only between 2 groups
(no edges within a group)

Question: is there an algorithm that colors with the fewest possible colors?

Answer: yes! consider every possible coloring, pick best one.

Problem: There are exponentially many colorings!

n^n possible colorings
(n choices of color for each of n verts)

Question: is there an efficient and optimal algorithm?

Answer: Nobody knows, probably no, if yes, it can be used to solve lots of other problems!

(e.g. decrypt all modern communication)

4-color theorem:

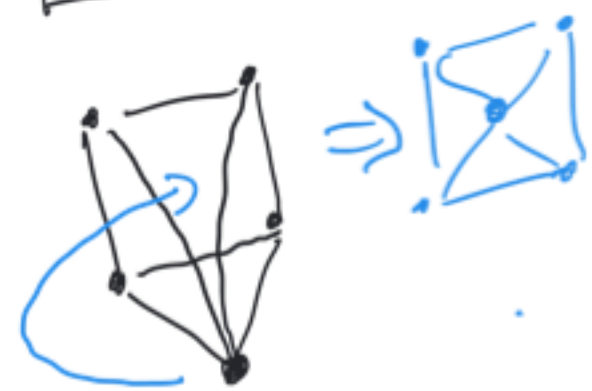
All planar graphs are 4-colorable
(i.e. with 4 colors).

Planar: Can be drawn in the plane without edges crossing:

planar

not planar:

planar:



Can move vertices!

Proof so complicated, needed to be constructed by a computer!