Lectures 21: Directed Graphs Pt.2

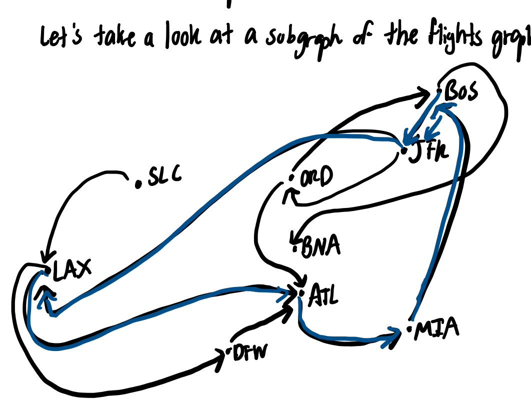
Wednesday, October 30, 2024 11:29 PM

Announcements:

1. Homework #11 due Tuesday (#5 axed)
2. My office hours after class

Review: Directed Graphs

Let's take a look at a subgraph of the flights graph



Vertices: The airports in the subgraph

Directed Edges: The flight routes between airports in the subgraph

Outdegree of JFK? 2

Indegree of LAX? 2

Sources? SLC

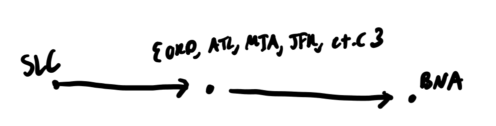
Sinks? BNA

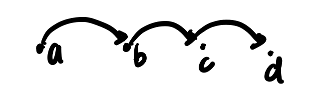
Path? SLC → LAX → ATL → MIA

Walk? BOS → JFK → LAX → ATL → MIA → BOS → JFK

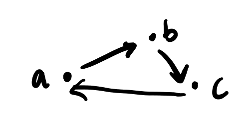
Cycles? BOS → JFK → LAX → ATL → MIA → BOS

SCCs? G^{SCC}?





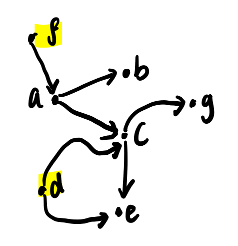
Lemma: G^{SCC} has no directed cycles.



Is the airport graph a DAG?

DAG = Directed Acyclic Graph ✗

Let's make a topological sort of a random graph:



f a d c e b g

f d a b c e g

_ _ _ _ _ _ _

- A topological sort of a DAG, is a sequence σ, in which every vertex appears once, such that for any edge $a \to b$ in the graph, the vertex a appears before b in the sequence

Prove the following:

1. The first vertex in a topo-sort is a source
2. The last vertex in a topo-sort is a sink
3. Only DAGs can have topo-sorts

1) Assume f.p.o.c that first vertex v is Not a Source:

This means that there is some a such that

$a \to v$

Contradiction! a cannot come after v in the topo-sort and thus must be the first vertex

2) Assume f.p.o.c that last vertex, b, is not a sink

There must be a vertex, v such that $b \to v$

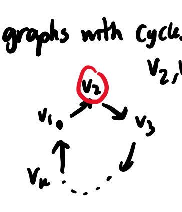
Thus, v cannot come before b in the topo-sort,

and v must be the last vertex, contradiction!

3) Assume f.p.o.c that directed graphs with cycles can have topo-sorts.

$V_2, V_3, V_4, \ldots V_k, V_1, V_2$

Consider the cycle, C,



Consider two vertices in that cycle V_a and V_b

Toposort ..... V_a .... V_b .....

V_{a-2} V_{a-1} V_{b+1} .. V_{b+2}

V_{a-i} and V_{b+i} will eventually be the same vertex, contradiction

Consider the following graph. What is the "algorithm" for finding a topological sort of a graph?

Step 1: Find source(s) and add to ordering

Step 2: Delete source(s)

Step 3: Repeat Step 1 until no vertices left

Prove every DAG has at least 1 topo sort

Induction on the # of vertices, n

BC: n=1 a $\underline{a}$ ✓

IH: A DAG with k vertices has a topo-sort

IS: W.T.S that DAG with k+1 vertices has a toposort

$v_1 \to v_2 \to v_3 \ldots v_k$; $v_2 \to v_{k+1} \ldots v_q$

Remove the sink vertex. (sink has to exist in a DAG)

I now have a DAG with k vertices

By IH, this DAG has a topological ordering: x

Add the sink back to the DAG, the new topological ordering will be x + sink

$\underbrace{\_\ \_\ \_\ \_\ \_\ \_}_{x}\ \underline{\text{sink}}$

/{v}

By IH, G′ has a valid topo-sort and our algorithm will find it = σ

v

k vertices

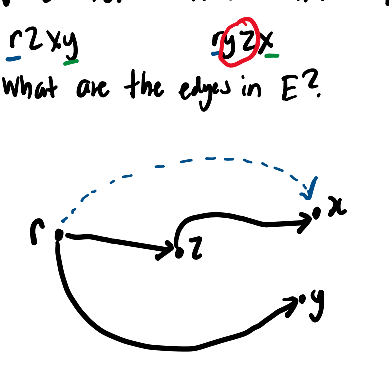
$\boxed{v + \sigma}$ ✓

Consider the directed graph (DAG), with nodes:

V = {r, x, y, z}. Here are all the topological sorts of G:

rzxy   ryzx   rzyx

What are the edges in E?   z → x



Graph Coloring:

There's n classes at GW

Each class needs a 2 hour exam slot for finals

Since each student may be in multiple classes, the exams need to be scheduled such that two courses that have students in common don't fall at the same time.

Question: What's the minimum number of timeslots required?

Vertices: Class [1...n]

Edges: (x, y) such that class x and class y share a student