CSCI 3100

Using flows

Today:

- HW is posted
  ↳ due Monday, Oct 30

Last week: max flow

Now: Applications

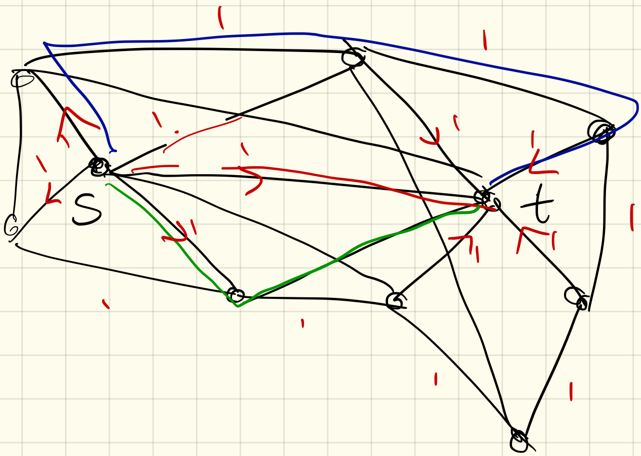
The real power of flows is how many problems can be solved using it!

Steps:

1. Model problem as a graph
2. Analyze runtime
3. Correctness:
   soln to problem $\iff$ flow of some value

Example: Edge disjoint paths

Goal: find the number of edge disjoint paths between s & t in G.



How?

Put capacity 1 on each edge

Calculate max flow f.

Outpute value(f).

Runtime:

runtime of F-F:

$O(mC)$ or $O(m|f|)$

(etc)

Correctness:

k edge disjoint s-t paths $\iff$ flow of value k

pf:

$\Rightarrow$: Spps I get k paths.

Since edge disjoint, I can push 1 flow along each path.

This respects capacity (since ≤ 1 on each edge) & vertex constraints, since s-t path $\to \circ \to$

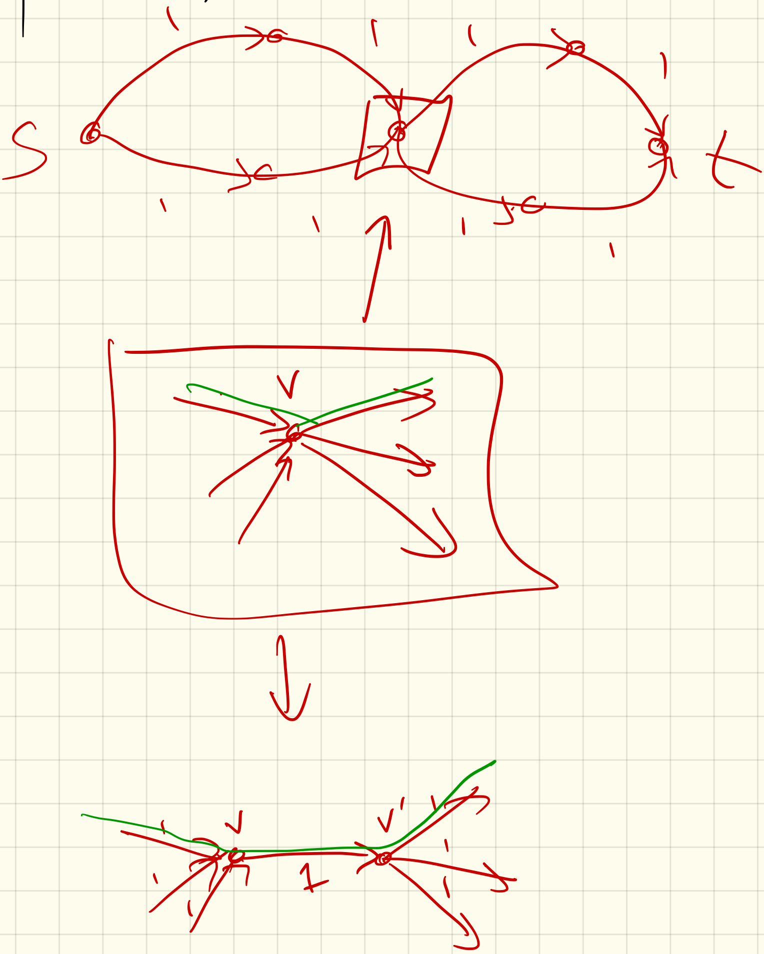
$\Leftarrow$: flow f of value k:

Get 1 path: pick edge of out s with $f(e)=1$.

Next vertex must have edge out w/ flow $= 1$: continue until you hit t $\to P$

Continue after remove those edges!

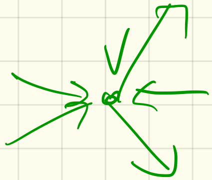
What about vertex disjoint paths?



Modify G:

Foreach $v \neq s, t$,
create 2 copies v', v'':

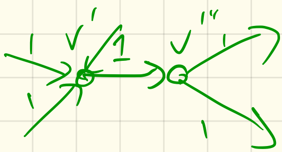
G



add directed edge of cap $= 1$ from v' to v''

any incoming edges to v becoming incoming to v'

G'



outgoing $\to v''$

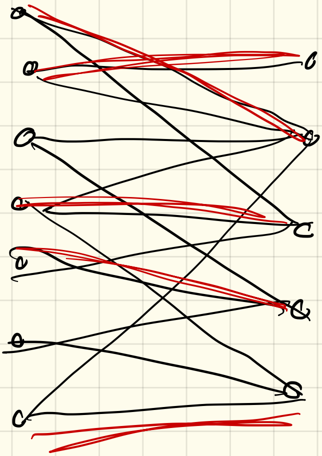
all capacity $= 1$

flow of value k in G'

$\iff$ k disjoint paths in G

Problem: Bipartite matching

set of edges s.t. each vertex is used ≤ 1



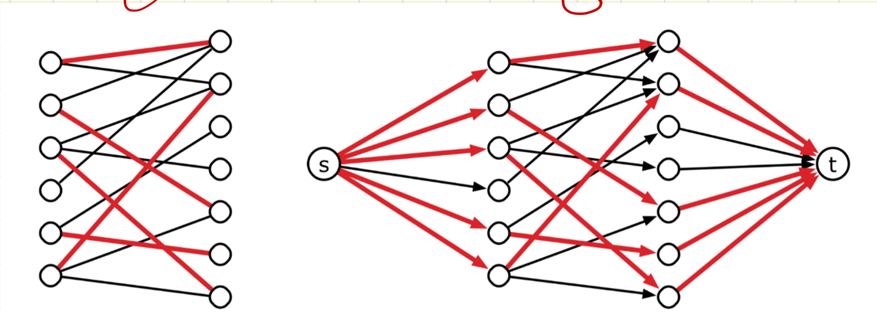
maximum matching

Why?

connect to all sorts of matching problems

How? Flows!

G G'



A maximum matching in a bipartite graph G, and the corresponding maximum flow in G'.

Alg:

Construct a new graph:

Add s & t.

Direct all edges from $L \to R$.

Send $s \to L$ edges & R to t edges

Runtime: Capacity 1 on all edges

Run flows on G'

Analyze in terms of G: $O(|E(G')| \times |f^*|) = O(m \cdot n)$

Correctness

matching w/ k edges
$\iff$ flow of value k

$\Rightarrow$ easy

$\Leftarrow$: Take flow:
s-t flow uses
only L-R edges,
& can ~~re~~decompose
into paths & hence
a matching

Another: Assignment Problems

Ex:
- n doctors at a hospital
- k vacation days

Need:
- A doctor scheduled on every vacation day
- no doctor scheduled on more than 3 vacation days
- each doctor submits a list of ≥ 5 vacation days that they are available to work on

Q: Is there a feasible schedule?

To Solve: build a graph

Runtime:

Correctness:

flow of size k

$\iff$ valid schedule