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: Solve to problem flow of some value
Example: Edge disjoint paths

Goal: find the number of edge disjoint paths between s and t in G.
How? Put capacity 1 on each edge. Calculate max flow. Output value $(f)$. 

Runtime: 

runtime of $F^F$: $O(mC)$ or $O(mf)$ (etc)
Correctness:

\[ k \text{V paths } \iff \text{flow of value } k \]

pf: edge disjoint s-t paths

\[ \Rightarrow \text{ Spps I get } k \text{ paths.} \]

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

This respects capacity (since \( \leq 1 \) on each edge) & vertex constraints since s-t path \[ \rightarrow \]

\[ \Leftarrow \text{ flow f of value } k. \]

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

Next vertex must have edge out \( v \) (flow = 1): continue until you hit t \[ \rightarrow P \]

Continue after remove those edges!
What about vertex disjoint paths?
Modify $G$:

For each $V 
eq s,t$, create 2 copies $v_i, v''_i$:

- $G' := G$
- add directed edge of cap=1 from $v'_i$ to $v''_i$
- any incoming edges to $v'_i$ becoming outgoing to $v''_i$

all capacity = 1

flow of value $k$ in $G'$

$\Rightarrow k$ disjoint paths in $G$
Problem: Bipartite matching

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

Why?

Connect to all sorts of matching problems
How? Flows!

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

Algorithm:

Construct a new graph:
- Add $s$ and $t$.
- Direct all edges from $L \to R$.
- Send $s \to L$ edges $\leftrightarrow R$ to $+t$ edges.

Runtime: Capacity $1$ on all edges.
Run flows on $G'$.
Analyze in terms of $G'$:
$$O\left(\frac{|E(G)|}{k} \cdot \log^3 k\right) = O(m-n)$$
Correctness

matching w/ k edges

⇐ flow of value k

to easy

⇐: Take flow:
s-t flow uses only L-R edges, can decompose into paths 

⇒ a matching
Another: Assignment Problems
Ex: on doctors at a hospital
  ok 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 \( \geq 5 \) 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