Edge-Disjoint Paths

Given directed graph $G = (V, E)$ and two vertices $s, t$.

Want a max size set of $(s, t)$-paths that don't share any edges.
Alg:
Give each edge a capacity of 1.
Compute a max $(s,t)$-flow.

Claim: There exists a flow $f$ s.t. $\|f\| = k$.

$f(e) = 1$ if $e$ is in some path of $P$.
$f(e) = 0$ o.w.
Claim: There is a set of 188 edge-disjoint paths.

Each path has weight 1. So there are 188 of them.
They are edge-disjoint. Othewise, we'd have flow \( \geq 2 \) on an edge.

So if \( k^* \) is max # paths \( |\varphi^*| \geq k^* \), we can grab \( k^* \) paths from it, so

\[ k^* \geq |\varphi^*| \]

\[ \Rightarrow \quad k^* = |\varphi^*| \]

Time: Orlin: \( O(VE) + O(k^*E) \)

If \( G \) is simple, \( |\varphi^*| \leq |V|-1 \)
So just use Ford-Fulkerson in $O(\mathrm{E}1^{\mathrm{E}1}) = O(VE)$ \[\frac{1}{2} O(k^*E)\]
Bibartite Matching

Given an undirected bipartite graph

\( G = (L \cup R, E) \)

(so no L-L or R-R edges)

A matching \( M \subseteq E \) has no two edges of \( M \) sharing an endpoint. (i.e. a subgraph where all
vertices have degree $D$

Maybe $L$ is mol students $+ R$ is internships.

Goal: Find a matching of max size.

Alg: Create $G'$ from $G$ by:
1) orient edges from $L$ to $R$
2) add vertices $s + t$
3) edge $s \Rightarrow v$ for all $v \in L$
4) add edge $u \Rightarrow t$ for all $u \in R$
5) Assign capacity 1 to all edges.

Claim: For any matching $M$ in $G$, there is a feasible $(s, t)$-flow $f$ in $G'$ such that $|f_M| = |M|$

For each edge $uv \in M$, set $f(s \rightarrow u) = f(u \rightarrow v) = f(v \rightarrow t) = 1$
Any edges we missed got flow 0.

\( f^* \): max flow assumed integral

\( M^* \): max matching

we know \( |f^*| \geq |M^*| \)

Claim: There is a matching \( M' \) s.t. \( |M'| = |f^*| \).

Take all \( uv \in E \) s.t.

\( f^*(u \to v) = 1 \).

Time with FF: \( O(|f^*|E) = O(VE) \)
Could change \( A \) times we use the edges or vertices by changing capacities.

Assignment Problems
Project Selection

Given a directed acyclic graph $G = (V, E)$,

$V$: the projects

depend on each other whenever $u$ depends on $v$

$\mathcal{R} \subseteq V \times R$

$\mathcal{R}(v)$ profit if we complete $v$

$\mathcal{R}(v)$ negative profit if $\mathcal{R}(v)$ cost.
Goal: Select $X \subseteq U$ that includes dependencies for all $x \in X$.

Want to max $\max_{x \in X} \delta(x)$.
Think of finding a cut $(S, T)$ of $G$. Turned down projects.

Turn $G$ into a flow network $G'$.

1) Add source $s$ and target $t$.
2) For each $v$ st. $\delta(v) > 0$, add $s \rightarrow v$.
   For each $u$ st. $\delta(u) < 0$, add $u \rightarrow t$. 
3) \( c(s \to v) = \mathcal{A}(v) \quad \forall \mathcal{A}(v) > 0 \)
\( c(u \to t) = -\mathcal{B}(u) \quad \forall \mathcal{B}(u) < 0 \)
\( c(u \to v) = \infty \quad \forall u \to v \in G \)

Let \((S, T)\) be any finite capacity \((s, t)\) cut, we won't have \( u \in S, v \in T \) if \( u \to v \in G \), so

\( X : S-s \) is a feasible solution to our problem.
\[ P := \exists_{v} \max \{ 0, \delta(v) \} = \exists_{x} \delta(x) \]

For any \( x \in V \), let

\[ \text{cost}(x) := \exists_{u \in X} \cdot \delta(u) = \exists_{u \in X} c(u \rightarrow x) \]

\[ \text{yield}(x) := \exists_{v \in X} \cdot \delta(v) = \exists_{v \in X} c(s \rightarrow v) \]
\[ \text{profit}(x) = \exists y(x) = \text{yield}(x) - \text{cost}(x) \]

By def., \( P = \text{yield}(V) = \text{yield}(S) + \text{yield}(T) \)

Assuming \( II(S, T) \) is finite
\[ II(S, T) = \text{cost}(S) + \text{yield}(T) \]

profit(s) = \text{yield}(S) - \text{cost}(S) = P - \text{yield}(T) - \text{cost}(S)
= P - II(S, T) \]
So compute min cut to maximize profit.

Time: $O(VE)$