CPSC 420 Lecture 10: Today's announcements:

- HW2 is on Gradescope, due Feb 9, 23:59
- Examlet 2 on Feb 17 in class. Closed book & no notes
- Reading: Maximum Flows & Minimum Cuts [Algorithms by Erickson Ch. 10]

Today's Plan

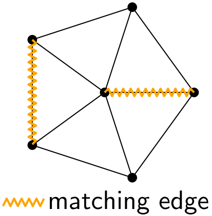
- Network Flow
  - Maximum matching in bipartite graphs
  - Pennant Race Problem
  - Open Pit Mining

Maximum Matching in Bipartite Graphs

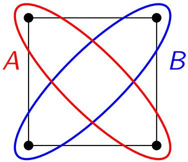
A **matching** in a graph G is a subset M of its edges with no vertex the endpoint of more than one edge in M.

A **maximum matching** is a matching with the maximum number of edges.

A **maximal matching** is a matching to which another edge cannot be added to form a new matching.



A **bipartite graph** is a graph $G = (V, E)$ where V can be partitioned into A and B such that $\forall (u, v) \in E$, either $u \in A$ and $v \in B$ or $u \in B$ and $v \in A$.



<u>Given</u> bipartite graph $G = (V, E)$ with partitions A and B
<u>find</u> maximum matching in G.

Maximum Matching in Bipartite Graphs Algorithm

Given bipartite graph $G = (V, E)$ with partitions A and B:

1. Create a flow network $F = (V', E')$

   $V' = V \cup \{s, t\}$ add source and sink

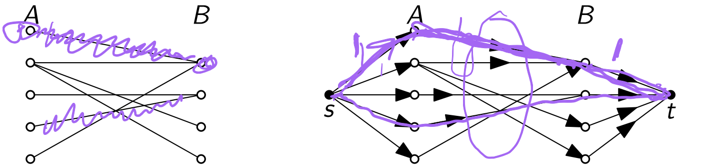
   $E' = E \cup \{(s, u) | u \in A\} \cup \{(v, t) | v \in B\}$

   Set all capacities to 1.

   NO

   1

2. Find maximum flow f in F
3. Output edges $(u, v) \in E$ such that $f(u, v) = 1$



Claim If M is a matching in G then $\exists$ flow f in F with $\text{size}(f) = |M|$.

$\text{size}(f^*) \geq |M^*|$

Claim If f is an integer-valued flow in F then there exists a matching M in G with $|M| = \text{size}(f)$.

$|M^*| \geq \text{size}(f^*)$

Pennant Race Problem

A B C D E

3 ~~1~~ 2 3 3

2 ~~2~~

3 3 3

B,C

B D

B E

Input	Example				
Given team A (your favorite team)	A				
list of teams $T_1, T_2, \ldots, T_n$	A	T_1	T_2	T_3	T_4
#wins for each team this season	~~3~~ 6	4	6	5	4
list of games remaining to be played					

$\cancel{(A, T_1)}, \cancel{(A, T_3)}, \cancel{(A, T_4)}, \underline{(T_1, T_3), (T_2, T_4), (T_1, T_2), (T_2, T_3)}$

Determine if it is possible for team A to win at least as many games as any other team by the end of the season.

1. Assume A wins all remaining games (it's possible)
   This removes some games.
2. Let w be number of A's wins. Let w_i be number of T_i's wins.
3. If $w < w_i$ for some i then return NO.
4. Create a flow network

Pennant Race Problem

Input

Given team A (your favorite team)

list of teams $T_1, T_2, \ldots, T_n$

#wins for each team this season

list of games remaining to be played

Example

A

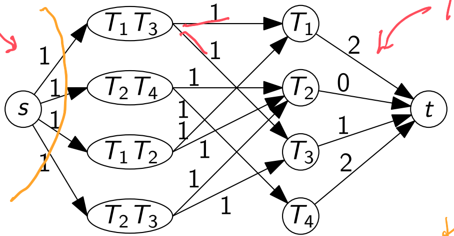
A	T_1	T_2	T_3	T_4
6	4	6	5	4

$(T_1, T_3), (T_2, T_4), (T_1, T_2), (T_2, T_3)$

Determine if it is possible for team A to win at least as many games as any other team by the end of the season.

limit, #games to below 6

flow out of each game to ≤ 1



if maxflow < #games

then NO HOPE

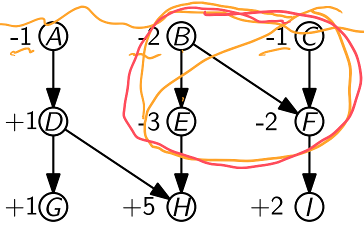
A has hope iff max flow size $=$ # games

Open Pit Mining

Imagine the earth is a lattice of cubes.
Every cube has a value (think “gold” minus “cost” to process)
Constraint: must remove some cubes before others (think cave-in)
Input: Directed acyclic graph $G = (V, E)$. $V =$ set of cubes
$E = \{(u, v) | u$ must be removed before $v\}$
$w(v) =$ value of cube v

GOOD

NOT GOOD



initial set

no incoming edges from outside set

Find most profitable set of cubes to process but obey constraints.

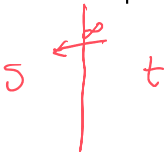
Maximum Value Initial Set

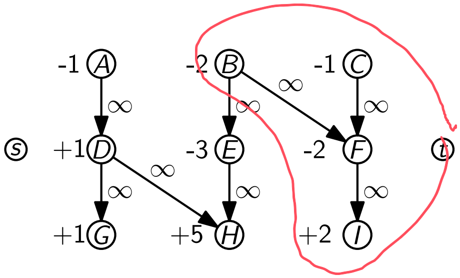
Convert the vertex-valued directed graph $G = (V, E)$ into a flow network so that

A. Any **finite capacity cut** corresponds to an **initial set**.

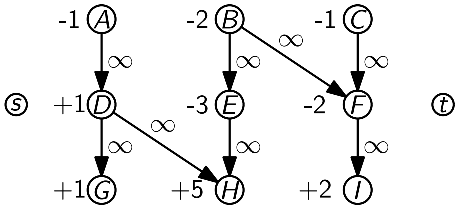
B. A **min capacity cut** corresponds to a max value initial set.

1. Add source s and sink t
2. Set capacity $c(u, v) = \infty$ for $(u, v) \in E$
3. Create an edge (s, v) or (v, t) for every $v \in V$ with finite capacity (to be determined)





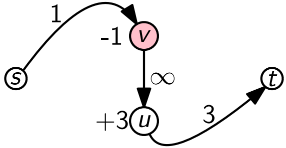
Maximum Value Initial Set



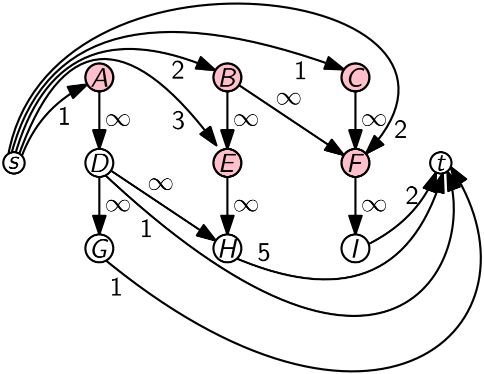
Claim: In this network any finite capacity cut (S, T) defines an initial set $T - t$ (and vice-versa).
Proof: If cut (S, T) has finite capacity then no original edges are directed into T from S so $T - t$ is an initial set.
If $U \subseteq V$ is initial then cut $(S = (V - U) + s, T = U + t)$ has finite capacity. Only edges $(s, u) | u \in T$ and $(v, t) | v \in S$ cross the cut from S to T (and they have finite capacity). $\square$

Maximum Value Initial Set

Idea If $w(u) > 0$, increase cut capacity if we **don't** take u ($u \notin T$).
If $w(v) < 0$, increase cut capacity if we **do** take v ($v \in T$).



Final flow network



Maximum Value Initial Set

For any initial set U, the capacity of the corresponding cut $(S = (V - U) + s, T = U + t)$ is

$$c(S, T) = \sum_{\substack{u \notin U \\ w(u) > 0}} w(u) + \sum_{\substack{v \in U \\ w(v) < 0}} -w(v)$$

$$\text{profit} \;=\; \sum_{\substack{u \in U \\ w(u) > 0}} w(u) + \sum_{\substack{v \in U \\ w(v) < 0}} w(v)$$

To maximize profit, minimize (over cuts (S, T), which define U)

$$\sum_{\substack{u \in U \\ w(u) > 0}} w(u) - \text{profit} \;=\; \sum_{\substack{u \notin U \\ w(u) > 0}} w(u) - \sum_{\substack{v \in U \\ w(v) < 0}} w(v) = c(S, T)$$