22 Mar 2021 Bipartite Matching, Etc.

Announcement:

Midterm TA evaluations due a week from today. Please give feedback to TAs, especially if you interact via office hours, regrades, ...

3 most important things about max flow.

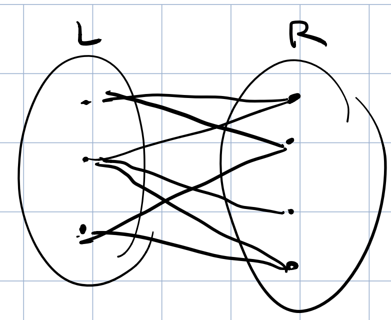
1) We can solve it in poly time.
   E.g. push-relabel $O(n^3)$.

2) Max-flow min-cut theorem.
   Maximum value of s-t flow
   = minimum capacity of s-t cut.

3) Flow integrality theorem.
   If all capacities are integers, then $\exists$ an integer valued maximum flow.
   (The output of Ford-Fulkerson or Push-Relabel is integer valued.)

Maximum bipartite matching

Given bipartite graph $G = (V,E)$
   bipartite means $V = L \cup R$ (L, R disjoint)
   and every edge has one endpoint in L, one in R.



Find a maximum matching: set of as many edges as possible with no vertex belonging to more than one edge.

Ex. L = TAs, R = time slots for weekly office hours.

edge = "this TA is available at that time"

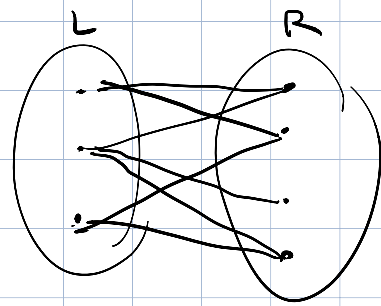
Max matching: If each TA can hold office hours once/week and no two office hours can be simultaneous schedule max # of office hours.

Ex. L = Black squares in a subset of chessboard

R = White squares

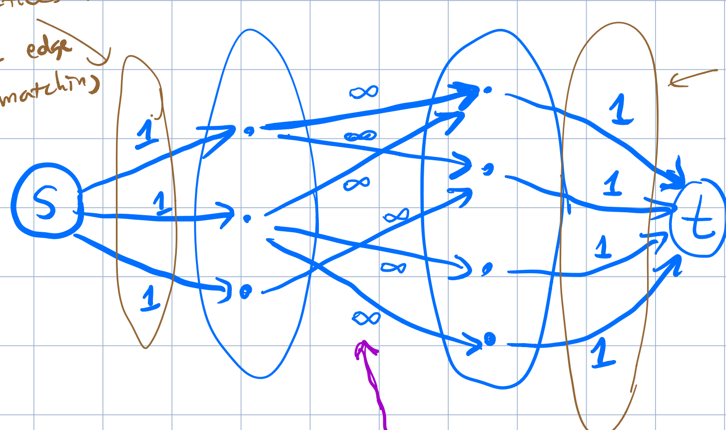
edge = adjacency

Max matching: what's the max # of disjoint dominoes we can place on the subset.



Max Bipartite Matching	Max Flow
Undirected	✓ Directed
No capacities	Capacities
No source, sink	✓ s,t must be defined
Pack edges into a graph without exceeding capacity of any vertex	✓ Pack paths into a graph without exceeding capacity of any edge.

This models the constraint that vertices in L belong to ≤ 1 edge of the matching



vertices in R belong to ≤ 1 matching edge

Everything you learned last week about solving network flow (+ max flow min cut)

(+ flow integrality) remains true if

- edges may have capacity ∞
- no infinite-capacity paths from s to t.

E.g. because the symbol "∞" here could refer to a finite integer greater than the combined capacity of all edges leaving s.

Recipe for reducing bipartite matching to max flow.

1. Build the flow network diagrammed above.

   $O(m+n)$ time to create

   $n+2$ vertices

   $m+n$ edges

   $n =$ # vertices of bipartite graph $= |L| + |R|$

   flow network has

   $$\left.\begin{array}{l} |L| \text{ edges in 1st layer} \\ m \text{ edges in middle layer} \\ |R| \text{ edges in last layer} \end{array}\right\} \begin{array}{l} m+|L|+|R| \\ = m+n \end{array}$$

2. Solve the maximum flow problem to find an <u>integer</u> max flow.

   $O((n+2)^3) = O(n^3)$. $\leftarrow$ if using Push-Relabel

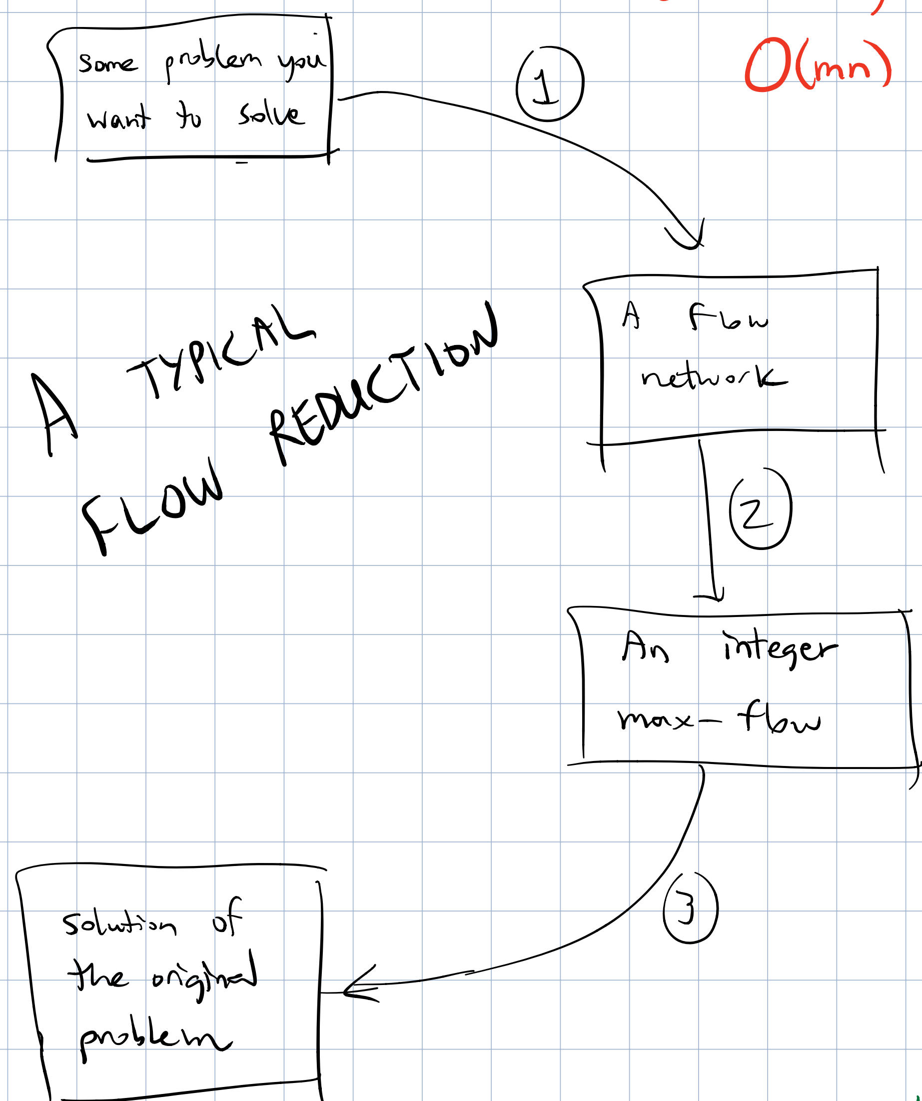
   $O(mC) = O(m \cdot |L|) = \boxed{O(mn)}$ $\leftarrow$ Ford-Fulkerson

3. Output $\{(u,v) \mid u \in L, v \in R, f(u,v) = 1\}$.

$O(m)$

Overall running time $O(mn)$

A TYPICAL FLOW REDUCTION



a.k.a. "bijection"

Justifying Correctness: establish a one-to-one correspondence between matchings and integer flows.

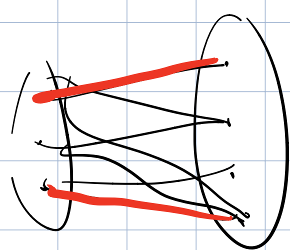
Matching $M \longmapsto$ flow

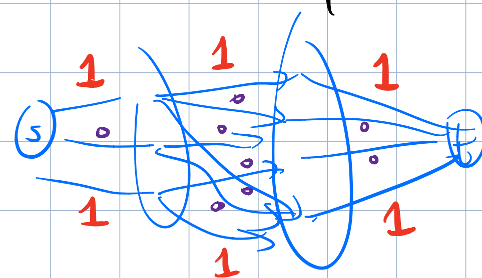
$$f(s,u) = \begin{cases} 1 & \text{if } u \text{ matched in } M \\ 0 & \text{if } u \text{ free} \end{cases}$$

Proof would need to show this f satisfies & cap cons constraints

$$f(u,v) = \begin{cases} 1 & \text{if } (u,v) \in M \\ \emptyset & \text{if } (u,v) \notin M \end{cases}$$

$$f(v,t) = \begin{cases} 1 & \text{if } v \text{ matched} \\ \emptyset & \text{if } v \text{ free} \end{cases}$$





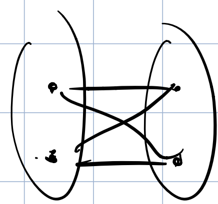
matching $\longleftarrow$ flow

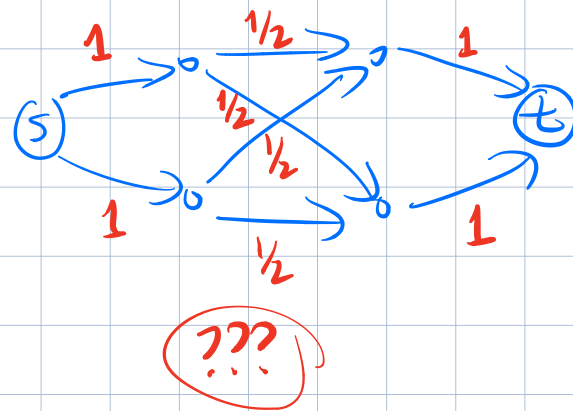
$$M = \{(u,v) \mid f(u,v) = 1\} \longleftarrow f$$

Proof would need to show when f is an integer flow this M is a matching.

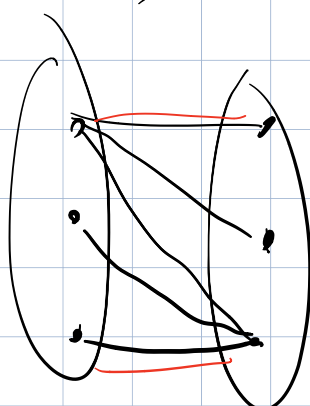
Under this bijection, size-of-matching equates to value-of-flow.

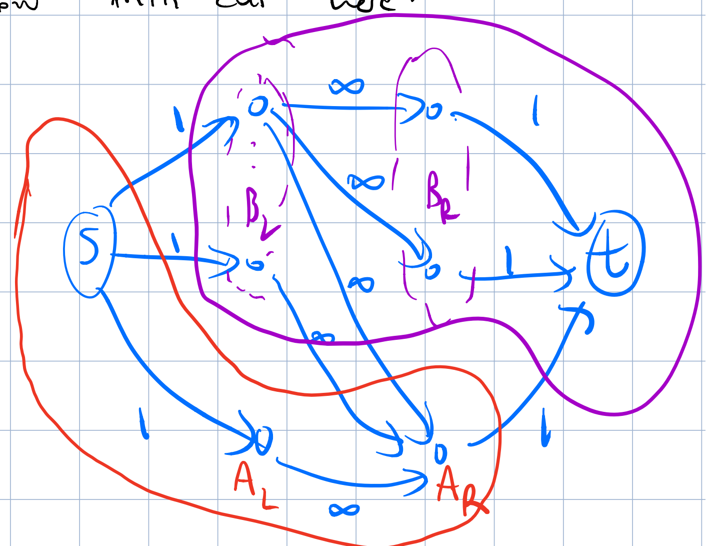
Why do we need an integer flow?





Interpreting max-flow min-cut here:





A finite capacity cut is (A, B) s.t.

$$A = \{s\} \cup \{A_L\} \cup \{A_R\}$$

$$B = \{t\} \cup \{B_L\} \cup \{B_R\}$$

s.t. $\forall$ edges u,v in middle layer,

it is not the case that $u \in A_L$, $v \in B_R$