12 March 2018 — Bipartite Matching

Given: bipartite graph with vertex set $V = L \cup R$ and edge set E s.t. every edge has one endpoint in each of L, R.

Note. It's easy (in linear time) to compute a partition of V into L & R satisfying the above, if one exists.

1. Choose any vertex. Assign it to L.
2. Continue doing BFS from that vertex, assigning alternating layers of the BFS tree to L and R.
3. If the graph has more than one connected component repeat steps 1&2 for every component.

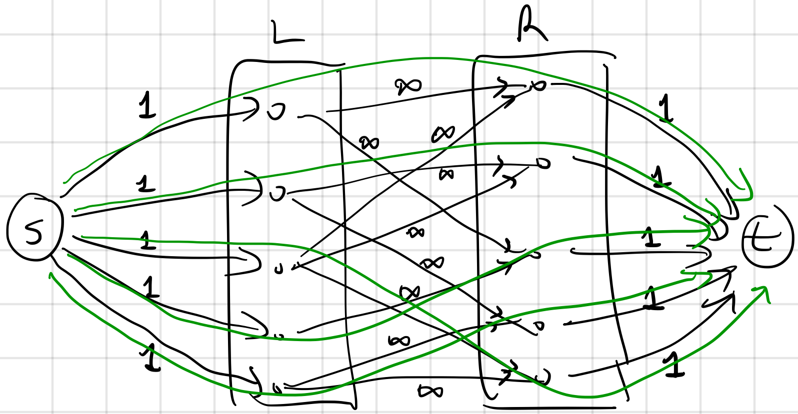
Reduction of bipartite matching to max flow:

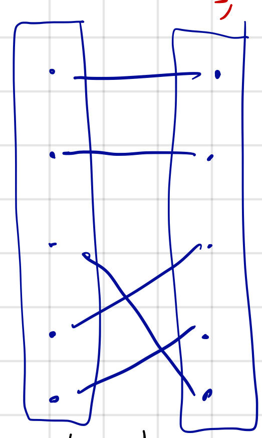
Construct a flow network with

$\boxed{\text{VERTICES}}$ - source s, sink t
- $L \cup R$

$\boxed{\text{EDGES}}$ - (s,u) $\forall u \in L$, $c(s,u) = 1$
- (v,t) $\forall v \in R$, $c(v,t) = 1$
- (u,v) $\forall u \in L, v \in R$, $(u,v) \in E$ $c(u,v) = \infty$

in this context ∞ denotes any integer greater than the combined capacity of edges leaving s, e.g. $\infty = |L| + 1$





Compute an integer-valued max flow in this network. (An int-valued max flow must exist because in a graph with integer capacities, Ford-Fulkerson outputs an integer valued flow.)

Output $M \triangleq \{(u,v) \mid f(u,v) = 1\}$.

Running Time. Define $n = |V|$, $m = |E|$ (in input graph).

Our flow network has $n+2$ vertices and $m+n$ edges.

The combined capacity leaving s, denoted C in the running time bound for Ford-Fulkerson, is equal to $|L| \le n$.

So running time $O(m+n)$ to build the network, $O((m+n)\cdot n)$ to find max flow, so $O(mn + n^2)$ overall.

If we preprocess input graph by removing isolated vertices, that takes $O(m+n)$ time, then in the new graph with $n' \le n$ vertices, $m \geq \frac{1}{2}n'$, running time $O(mn' + (n')^2)$ can be expressed as $O(mn') \le O(mn)$.

Running time for bipartite matching: $O(mn + n)$.

Correctness. Have to prove:
(a) M is a matching
(b) It has at least as many edges as any other matching.

There's a one-to-one correspondence

$$\{\text{Matchings in the bipartite graph}\} \longleftrightarrow \left\{\begin{matrix}\text{Integer-valued flows}\\ \text{in the flow network}\end{matrix}\right\}$$

$$M \longmapsto f_M$$

$$M_f \longleftarrow\!\!\dashv f$$

$v(f_M) = |M|$

$|M_f| = v(f)$

So f is a flow of maximum value $\Updownarrow$ M_f is a matching of maximum size.

$f_M :=$ flow that sends one unit on each path $s \to u \to v \to t$ for all edges $(u,v) \in M$

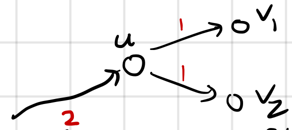
$M_f := \{ (u,v) \mid f(u,v) = 1,\ u \in L,\ v \in R \}$

Why does this work?

f_M satisfies conservation constraints by construction

satisfies capacity constraints b/c each vertex belongs ≤ 1 edge of M.

M_f is a matching because



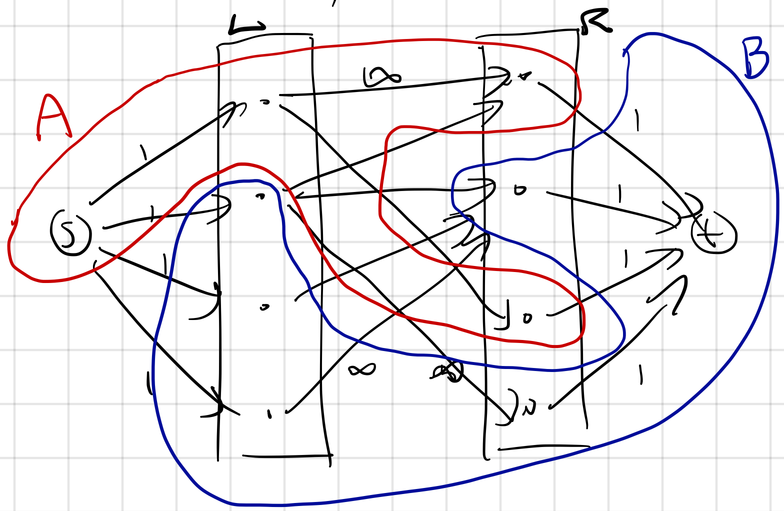
flow cons $\Rightarrow f(s,u) = 2$, capacity const $\Rightarrow f(s,u) \le 1$ Contradicting f is a flow!

Interpreting max-flow min-cut theorem in the bipartite matching problem.

$$\max\{|M| : M \text{ is a matching}\}$$

$$= \max\{v(f) : f \text{ is a flow in the network our reduction constructed}\}$$

$$= \min\{c(A,B) : (A,B) \text{ is a cut in the network}\}$$



If $c(A,B) < \infty$ then no infinite capacity edges may cross from A to B.

If $A_L := A \cap L$ then A must also contain every neighbor of A_L in R.

Let $A_L = A \cap L$, $A_R = A \cap R$, $B_L = B \cap L$, $B_R = B \cap R$.

- If $c(A,B) < \infty$ then A_R contains every neighbor of A_L and

$$c(A,B) = |B_L| + |A_R|$$

$$= |B_L| + |A_L| - (|A_L| - |A_R|)$$

$$= |L| - (|A_L| - |A_R|).$$

Max-flow min-cut theorem ensures that if the true max matching size is k, there must be a set A_L whose set of neighbors, A_R, is such that

$$|L| - (|A_L| - |A_R|) = k.$$

<u>Special case:</u> Bipartite G contains a matching that covers every vertex in L <u>if and only if</u> every subset $A_L \subseteq L$ has at least $|A_L|$ neighbors in R. ("Hall's Marriage Theorem").