Monday October 31 — Midterm 2

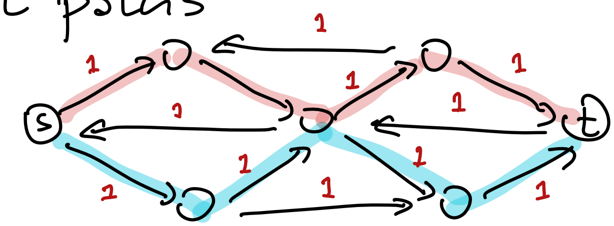
HW 4 5 6 7 + today!

Thursday — review session

Conflict — register by Fri

No problem 3

Edge-disjoint paths



Given directed $G=(V,E)$ vertices s, t

Find max # paths from s to t not sharing edges

Algorithm

Assign capacity 1 to every edge

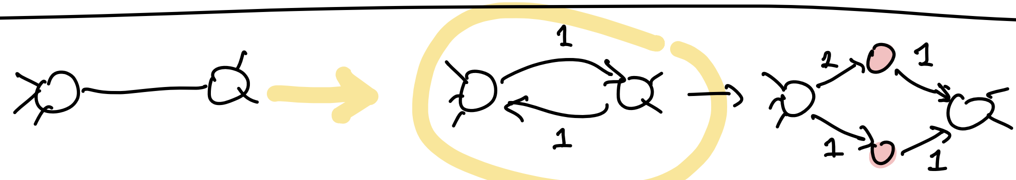
Compute max (s,t)-flow f^* ← Ford-Fulkerson $O(VE)$

Decompose f^* into paths ← $O(VE)$

$\boxed{O(VE) \text{ time}}$

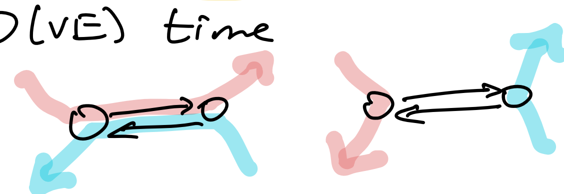
Peng et al 2022 $O(E^{1+o})$

Undirected?



still $O(VE)$ time

Vertex-disjoint?



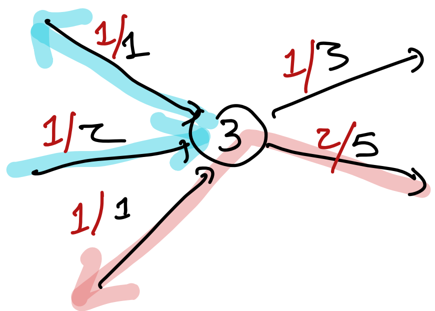
⇓

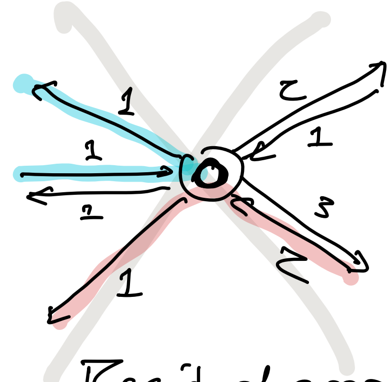
Vertex capacities (in addition to edge caps)

Feasible: $0 \le f(e) \le c(e)$

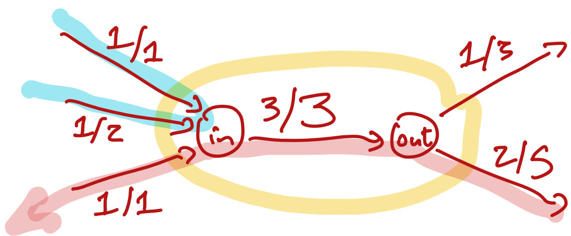
$+ \quad \sum_{u \to v} f(u \to v) \le c(v)$

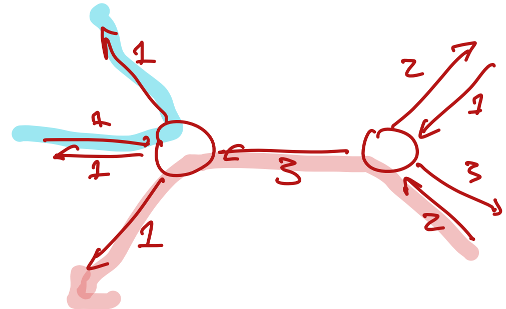
① ~~Change algorithm~~ ② Change graph

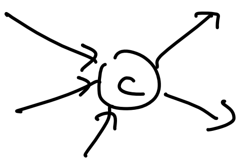


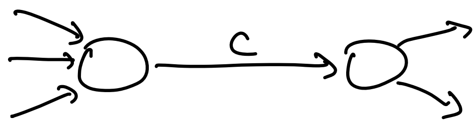


Residual graphs don't work!

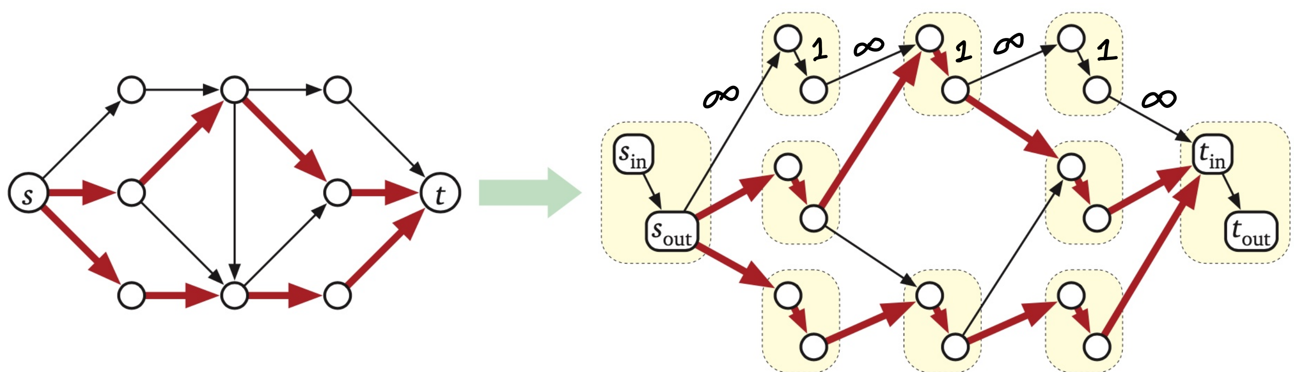








<u>Vertex-disjoint paths</u> <u>$O(VE)$ time</u>



More generally: Find max # paths

$\leq A$ through each edge

$\leq B$ through each vertex

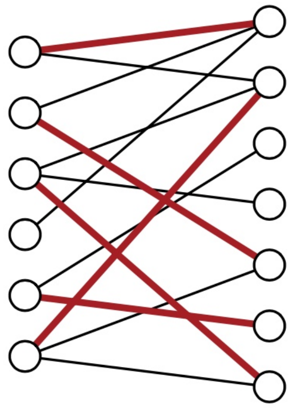
$O(VE)$ time

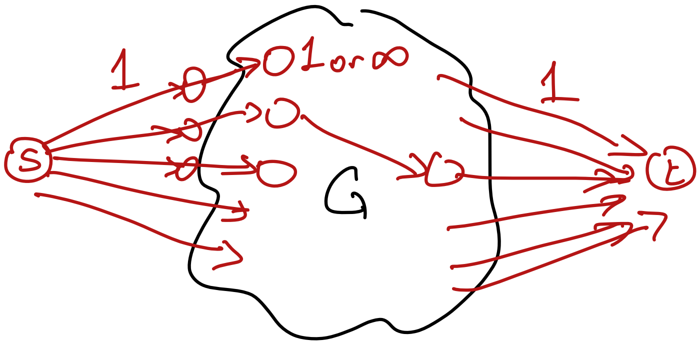
Given a bipartite graph

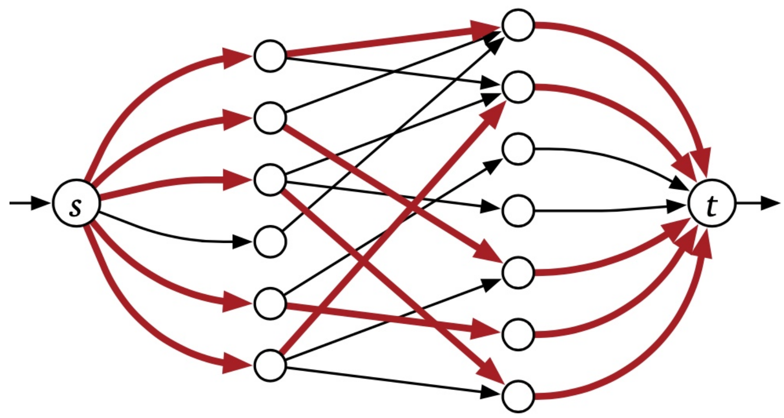
Find a maximum matching

= max # edges, no 2 sharing a vertex

bipartite graph







Add source s
edges $s \to L$

Add target t
edges $R \to t$

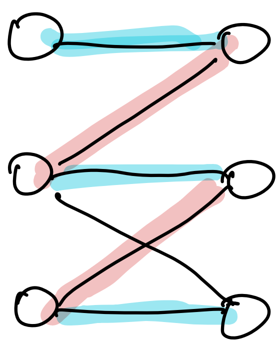
Direct $L \to R$

$c(e) \leftarrow 1 \quad \forall e$

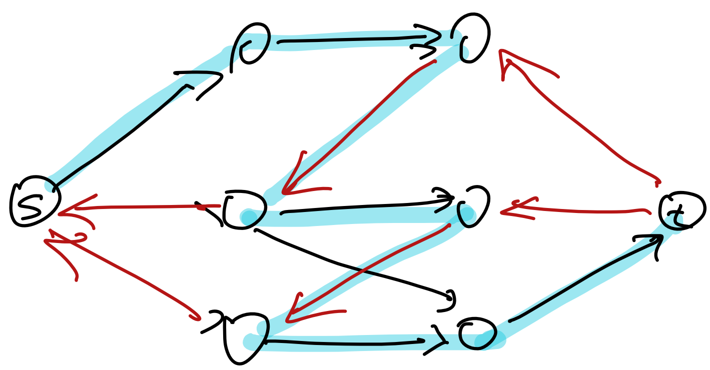
Max Flow → Flow Decomp

Integer Max Flow → Matching $= \{e \mid f(e) = 1\}$

$O(VE)$ time



Alternating path



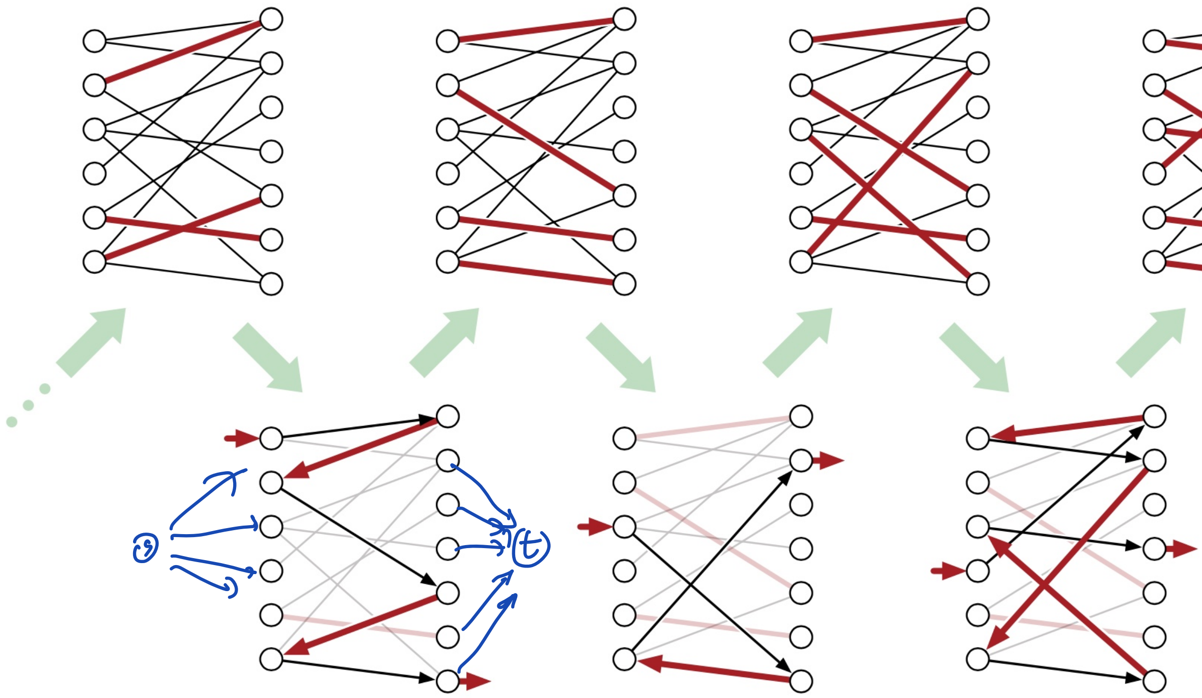
Augmenting path

MaxMatching(G)

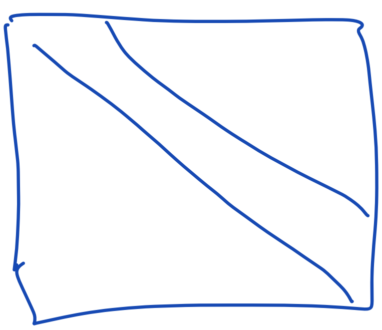
$M \leftarrow \emptyset$
while there is an alternating path in G
    $P \leftarrow$ any alt. path ← $O(E+V)$
    $M \leftarrow M \oplus P$ ← $O(V)$
return M

$O(V)$ iterations

$O(V^2 + EV)$



Berge (1957)
Jacobi (1836)



Permute rows + columns
all positive along diagonal