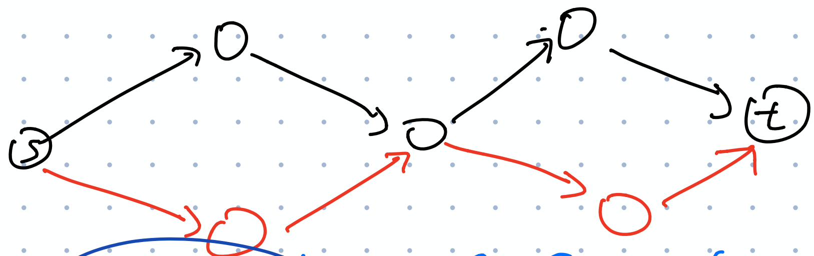
Edge-disjoint paths

no edges in common



Given a directed graph G, two vertices s and t

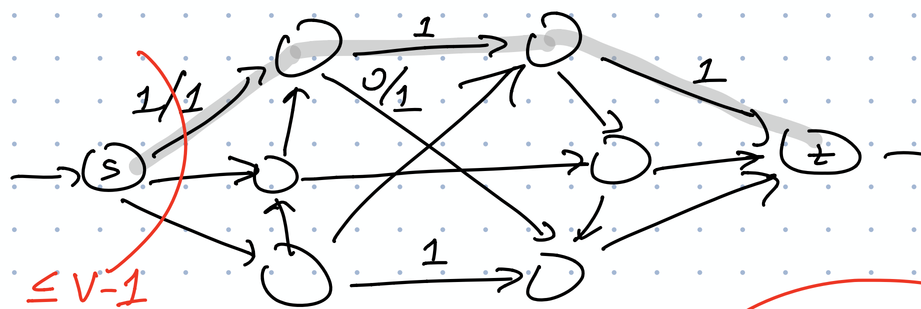
Find max # edge-disjoint paths from s to t

Give every edge in G capacity 1

Find integer max flow from s to t — Ford Fulkerson

Path decomposition (if we need paths)

(# paths = value of max flow)



Time:

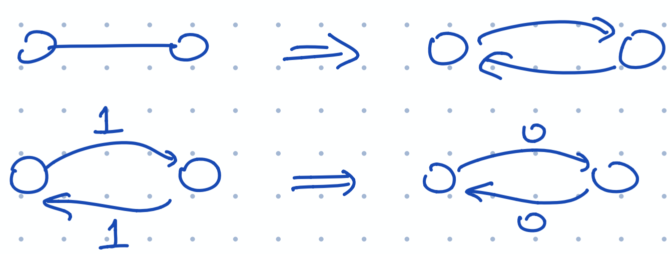
Ford Fulkerson $O(E \cdot |F^*|) = O(EV)$ time

$|F^*| < V$

Path Decomposition ~~$O(EV)$~~ time

$O(E)$

Undirected



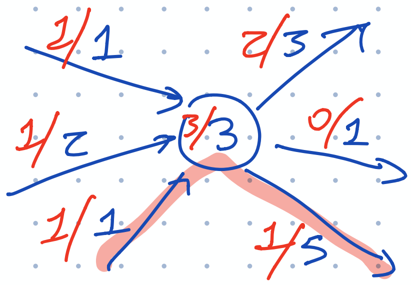
$+O(E)$

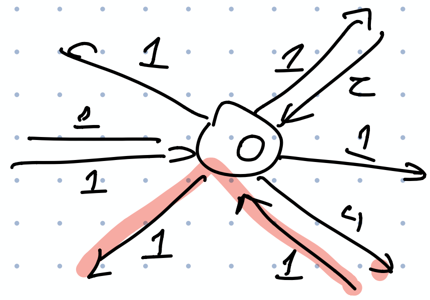
$\boxed{O(VE) \text{ time}}$

Vertex Capacities

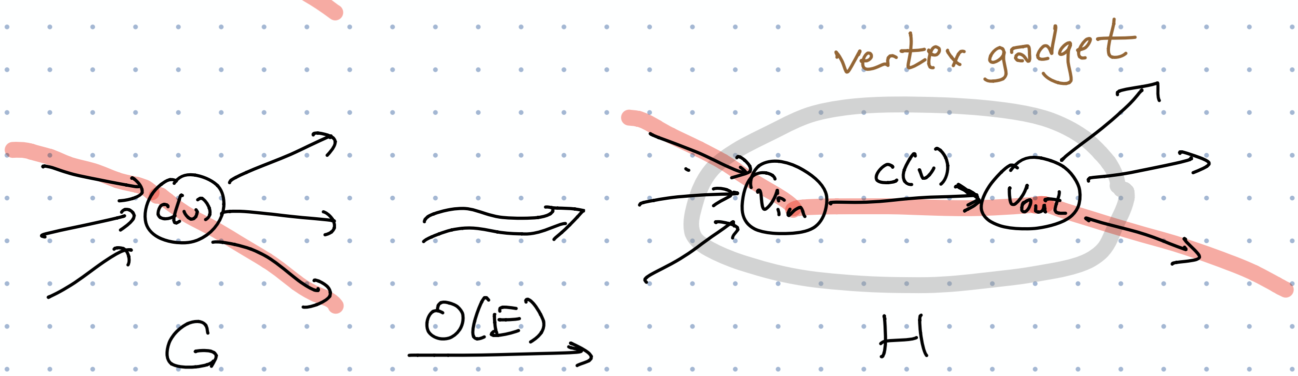
Feasible $\Longleftrightarrow$ $0 \le f(e) \le c(e)$ for all edges e

New variant: AND $\sum_{u\to v} f(u\to v) \le c(v)$ for all vertices v except s and t

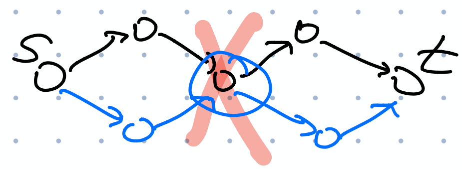




~~Change algorithm~~ or change graph?

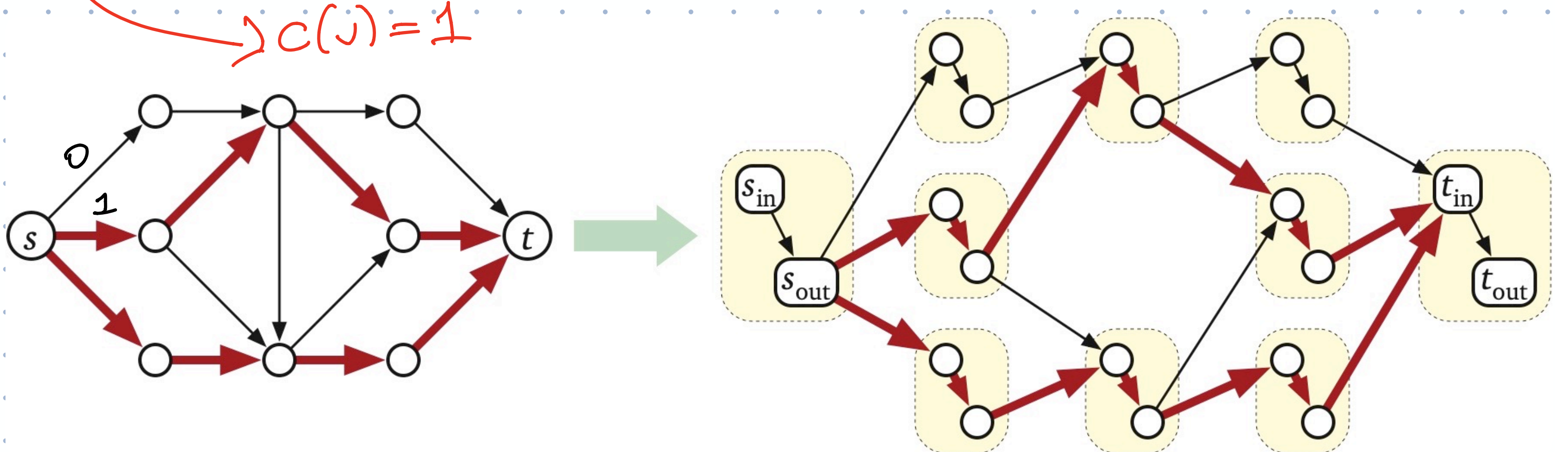


Vertex-disjoint paths



Max # vertex disjoint paths from s to t in G

$c(v) = 1$



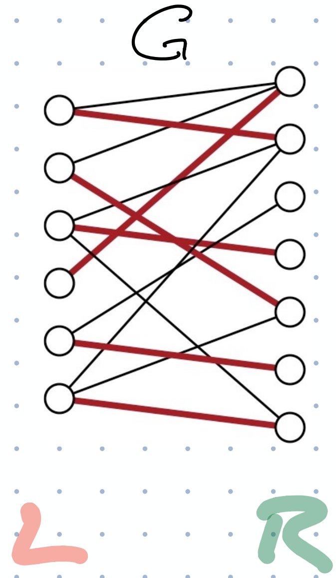
$\boxed{O(VE) \text{ time}}$

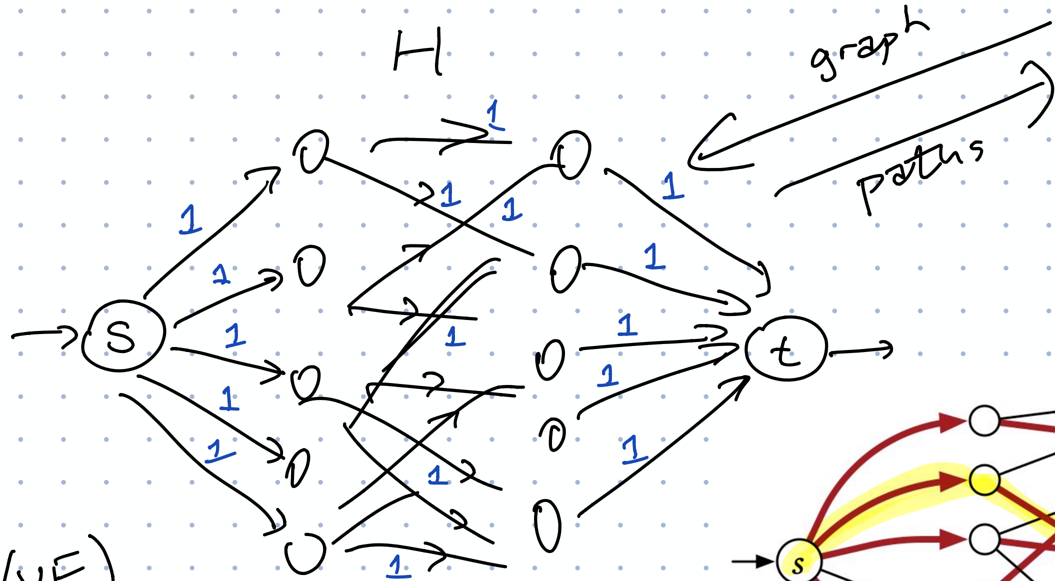
min # edge-disjoint paths from s_{out} to t_{in} in H.

Maximum Matching (Bipartite)

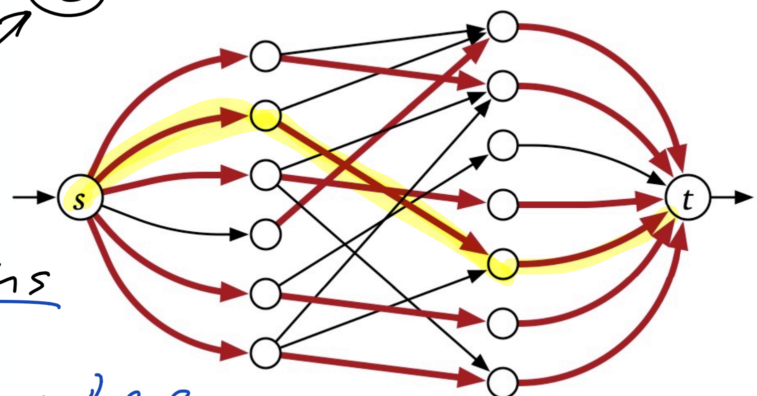
Given a bipartite graph $(L \cup R, E)$

Find maximum number of edges with no vertices in common





$O(VE)$

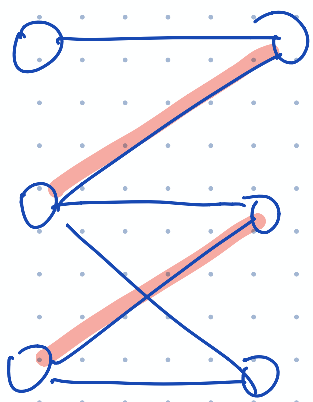


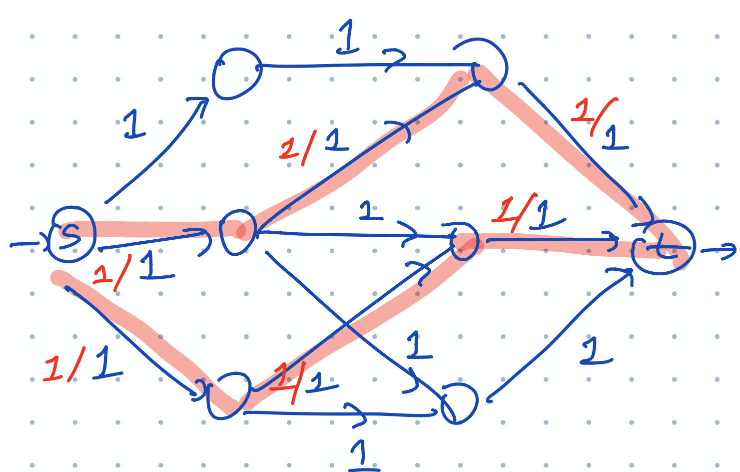
Find max # disjoint paths

||

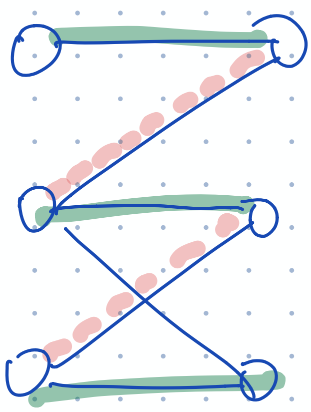
capacity $c(e)=1$ everywhere

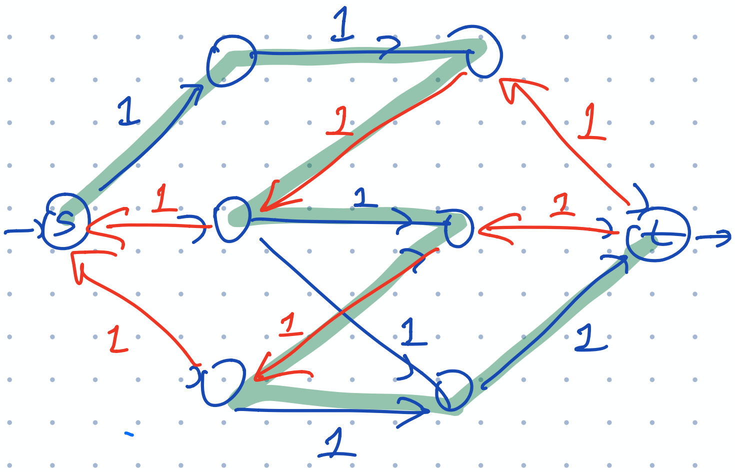
Compute integer max flow ← $O(VE)$ time FF

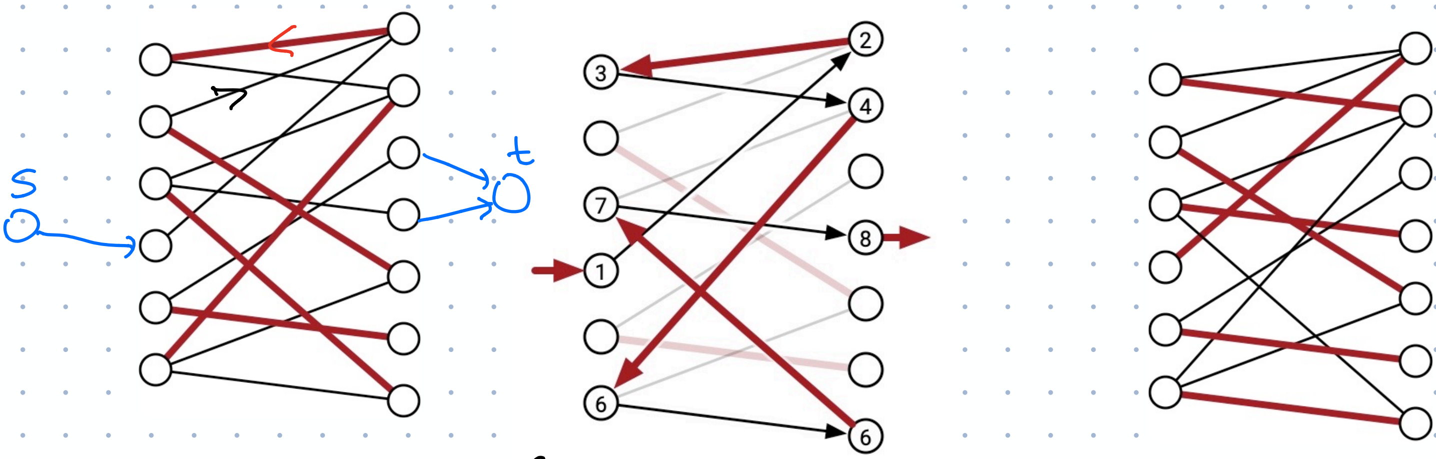




alternating path ← augmenting path







Prep: decompose G into components. For each component:

Max Matching(G)

M ← ∅

while there is an alt path in G ⟸ $O(\cancel{V+}E)$
setup + WFS

  P ← any alt path

  M ← M ⊕ P

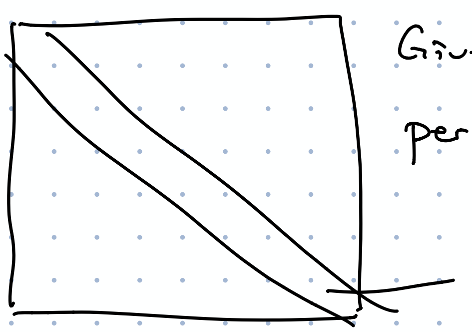
$O(V(V+E))$ time

$= O(\cancel{V^2}+VE)$ ~~(+V+E)~~

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

Berge (1957)

Jacobi (1836)



Given (0,1) matrix:

permute rows

max Σ diagonal