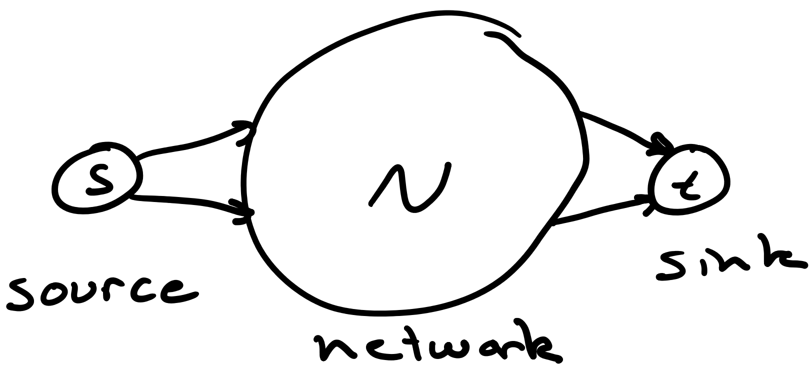
Flow networks

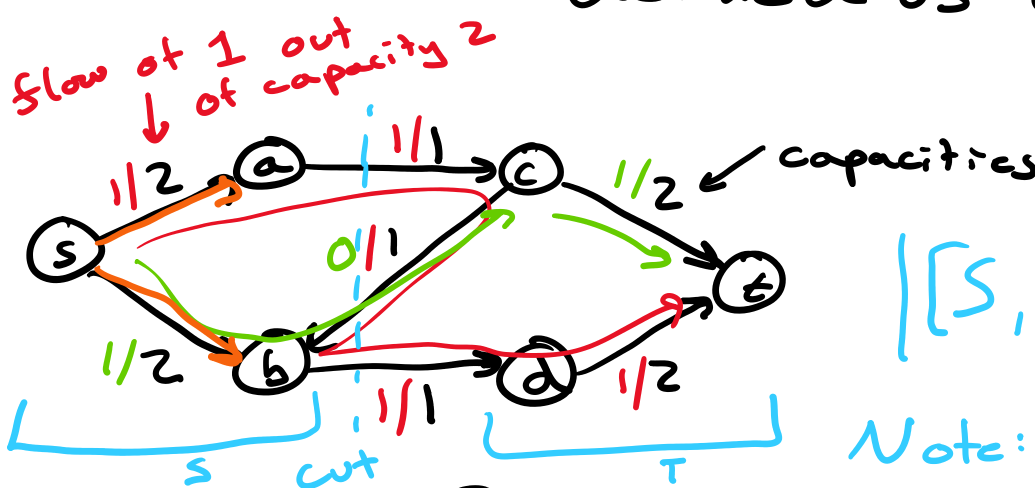


how much "flow" along each edge

$\forall e \in E(G)$: we have a <u>capacity</u>

defined as $c(e) \geq 0$

(integer)



$|[S, T]| = 2$

Note: flow = 2

a flow on G assigns to each

edge e some flow value $f(e)$

→ These flow values must be

<u>feasible</u>

$\forall e \in E(G): 0 \le f(e) \le c(e)$

$\forall v \in V(G)$: $f^-(v)$ = sum of flows on edges into v

$f^+(v)$ = sum of flows on edges out of v

$f^-(v) = f^+(v)$

total flow on network $\longrightarrow f^+(s) = f^-(t)$

Given feasible flow f,

we define <u>f-augmenting path</u> P_f

P_f goes from s to t

$\forall e \in E(P_f)$:

if P_f follows a forward direction then $f(e) < c(e)$

if P_f follows a backward direction then $f(e) > 0$

$\epsilon(e) = c(e) - f(e)$ = tolerance of e for forward edges

$\epsilon(e) = f(e) =$ tolerance of e for backward edges

Given P_f, we consider the minimum tolerance $\forall e \in E(P_f)$

$\rightarrow$ define it as z

To augment our flow:

$\forall e \in E(P_f)$: $f(e) \mathrel{+}= z$ for forward e

$f(e) \mathrel{-}= z$ for backward e

Source-sink cut

$$[S, T]$$

S = source set of vertices

T = sink set of vertices

Note: the size of $|[S,T]|$ is just the sum of capacities of the cut edges of the cut edges

$$= \sum c(e) \ \forall e \in [S, T]$$

$S = \{$ vertices that can be reached from s along a psuedo-f-aug. path$\}$

maximal path that starts at s but can't reach t

$T = \{V(G) - S\}$

Note: the size of any cut gives us a bound an maximum flow

$$|[S, T]| \geq val(f)$$

total flow on the network

Big Question

? ? ? ?

$\boxed{\text{Does minimum cut = maximum flow?}}$

A: yes

↳ Let's prove this via a few relations/equivalences

1. f is a maximum flow
2. no f-aug. paths on our network
3. $|[S,T]| = val(f)$

we'll show $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$

$(1 \Rightarrow 2)$ contrapositive

$\neg 2 \Rightarrow \neg 1$

there exists f-aug path $\Rightarrow$ f is not maximum

→ we've already demonstrated how to increase flow by using an f-aug path how to increase flow by using an f-aug path

(2 => 3)

no f-aug paths => $|[S,T]| = val(f)$

S = set of vertices from s that are reachable via psuedo-f-aug. paths

Note: $s \in S$, $t \notin S$

all edges from $S \to T$ have
$$c(e) = f(e)$$
all edges from $T \to S$ have
$$f(e) = 0$$

$$val(f) = \sum \text{flows from } S \to T - \underbrace{\sum \text{flows from } T \to S}_{=0}$$

$$val(f) = \sum \text{flows from } S \to T = \sum_{e \in [S,T]} c(e) = |[S,T]|$$

(3 => 1)

cut = flow => flow is maximum

Note: the capacities on edges gives us cut = flow

Q: Can we increase our flow?

A: No. Forward edges are at full capacity and backward edges are at zero flow

=> we cannot increase our flow

Combined with our earlier inequality

→ cut ≥ max flow

and cut = max flow

→ $\boxed{\text{minimum cut = maximum flow}}$ □

To get max-flow / min-cut (min s,t-cut)

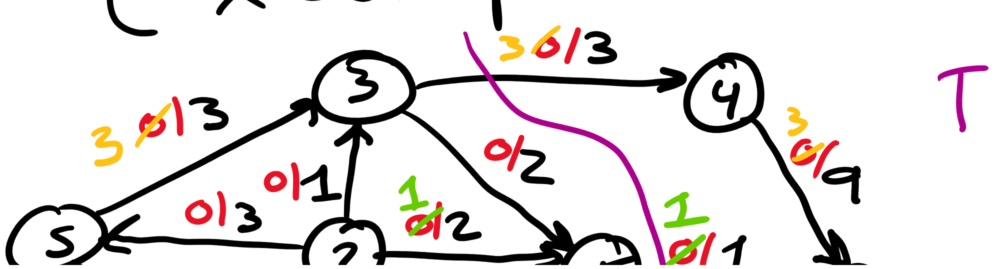
[Initialize all f(e) to zero
while ∃ some f-aug path P_f
  Find z = min tolerance on P_f
  update f(e): $\forall e \in E(P_f)$ with z
→ we're done

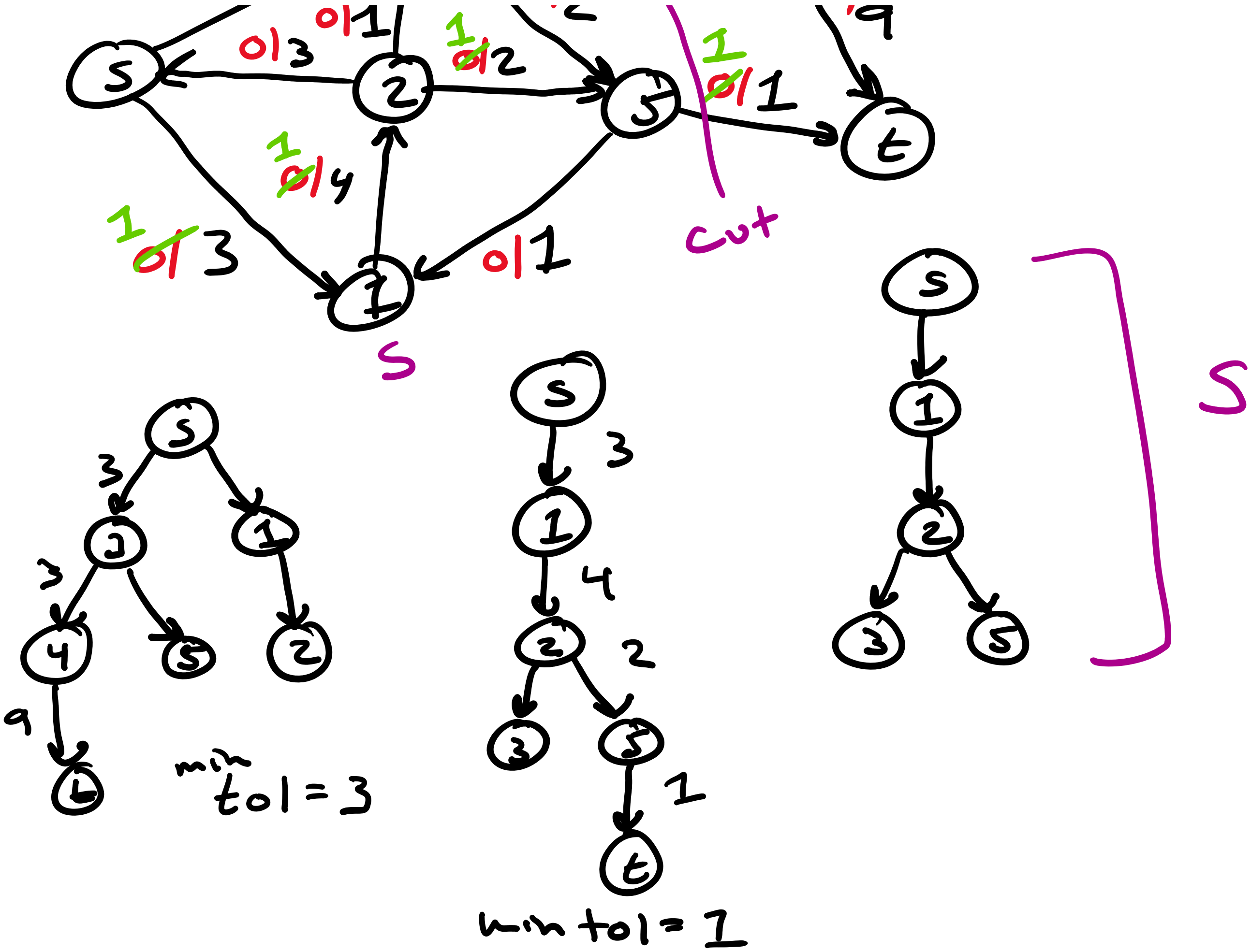
To get min cut:
  define our cut as [S, T]
  S = vertices reachable from s
    on psuedo-f-aug. paths
  T = everything else

Basic Algorithm:
  Ford-Fulkerson Algorithm

If we use BFS to find P_f:
  Edmonds-Karp Algorithm

Example





$$val(S) = 4$$

$$|[S,T]| = 4$$

$$\boxed{\text{max flow} = \text{min cut}}$$

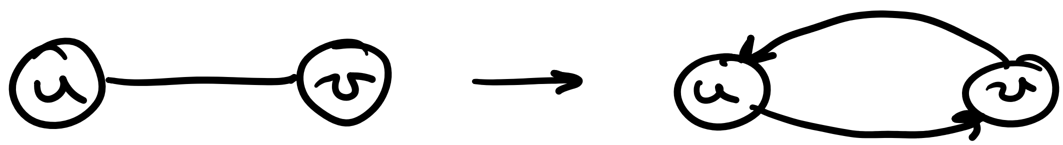
WP8

To make undirected G to directed G'

replace all $e = (u, v) \in E(G)$

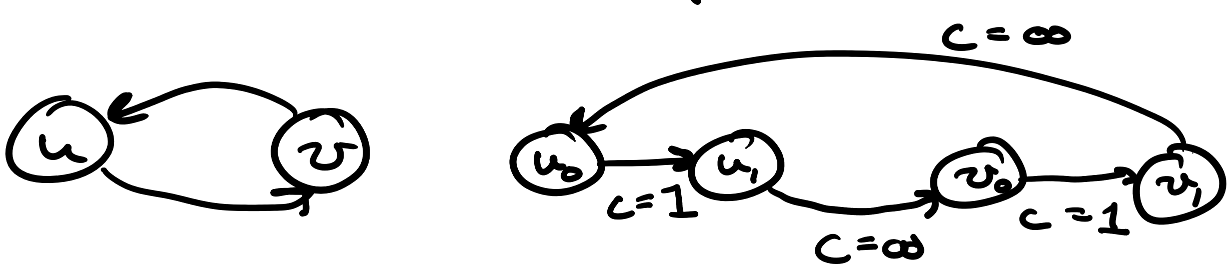
with $e' = (u \to v)$

$e'' = (v \to u)$



For problem 1:

* replace each vertex v with a single directed edge with unit capacity
* this edge goes from $v_0 \to v_1$ where v_0 has all of v's original incoming edges and v_1 has v's outgoing edges
* all other edges have infinite capacity



Note: we can define any pair of vertices as source/sink