Cormen 26.1-26.2 Flows

A network $N = (V, E, c)$
is a directed graph $D = (V, E)$ associated
with a capacity function $c : E \to \mathbb{R}_0$
$(c(u,v) \geq 0 \ \forall \ (u,v) \in A)$
If $(u,v) \notin E$ (not an edge of G)
then $c(u,v) = 0$

$\boxed{\text{Assumption in Cormen: if } (u,v) \in E \text{ then } (v,u) \notin E}$

NB: we give a more general definition of a flow than that in Cormen below.

A flow f in N is any function $f : E \to \mathbb{R}_0$
such that $0 \leq f(u,v) \leq c(u,v) \ \forall (u,v) \in E$

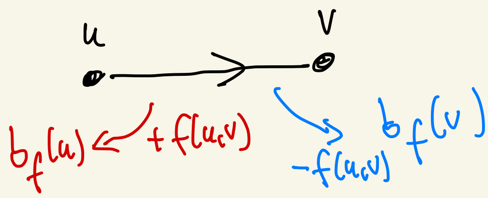
The balance b_f of a flow f is the function

$$b_f(v) = \sum_{(v,w) \in E} f(v,w) - \sum_{(u,v) \in E} f(u,v)$$

Observation For every flow f in a network $N=(V,E,c)$ we have $\sum_{v\in V} b_f(v) = 0$ (*)

Proof $b_f(v) = \sum_{(v,w)\in E} f(v,w) - \sum_{(u,v)\in E} f(u,v)$

So in $\sum_{v\in V} b_f(v)$ each edge (u,v) contributes $+f(u,v)$ in $b_f(u)$ and $-f(u,v)$ in $b_f(v)$ so 0 in total



Definition Let $N=(V,E,c)$ be a network and let $s,t\in V$ be distinct vertices. A flow f in N is an (s,t)-flow if there is some $K\geq 0$ so that

$$b_f(v) = \begin{cases} K & \text{if } v = s \\ -K & \text{if } v = t \\ 0 & \text{if } v \notin \{s,t\} \end{cases}$$

We say that an (s,t)-flow has flow conservation in all vertices distinct from s,t

The vertex s is the source and the vertex t is the sink

Definition The value of an (s,t)-flow f in $N=(V,E,C)$ is denoted $|f|$ and is defined to be

$$|f| = \sum_{v\in V} f(s,v) - \sum_{v \in V} f(v,s)$$

Note: This is the same $b_f(s)$ as $f(u,v)=0$ when $(u,v)\notin E$

so $|f| = b_f(s)$ and $|f| = -b_f(t)$

as $\sum_{v\in V} b_f(v) = 0$ by the observation (*)

Definition The maximum flow problem on $N=(V,E,C)$ with special vertices s,t is

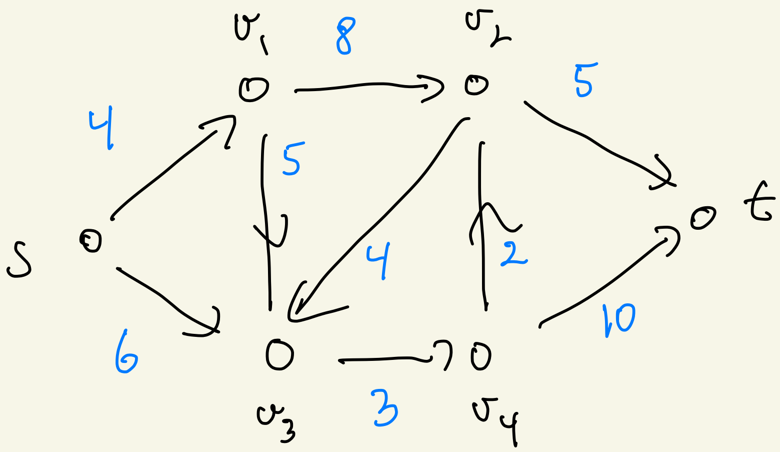
maximize K

such that

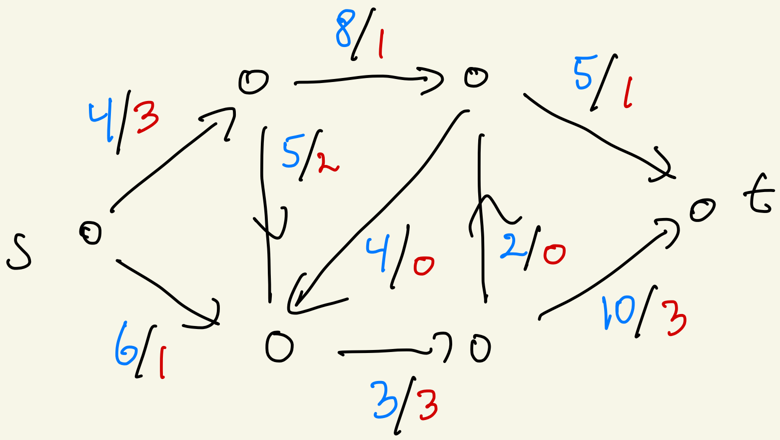
$$b_f(v) = \begin{cases} K & \text{if } v = s \\ -K & \text{if } v = t \\ 0 & \text{if } v \notin \{s,t\} \end{cases}$$

$$0 \le f(u,v) \le c(u,v) \quad \forall (u,v)$$

Example of (s,t)-flow:



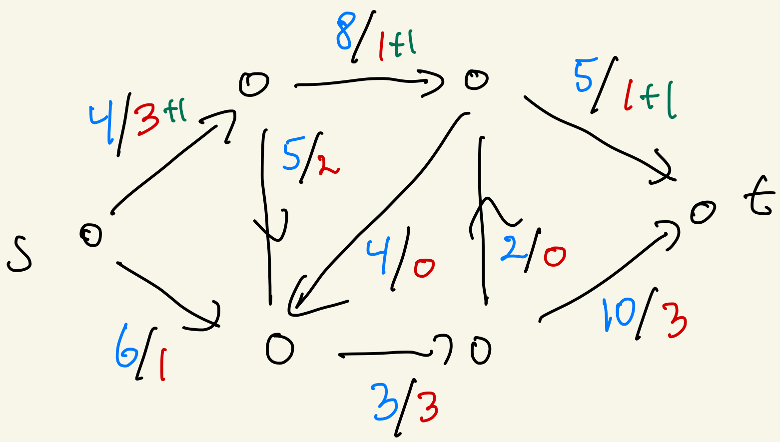
Network without flow



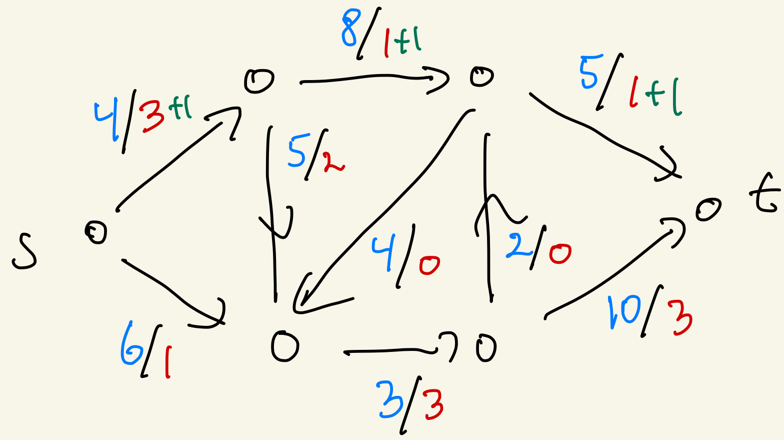
An (s,t) flow f of value 4

Is f a maximum (s,t)-flow?

No



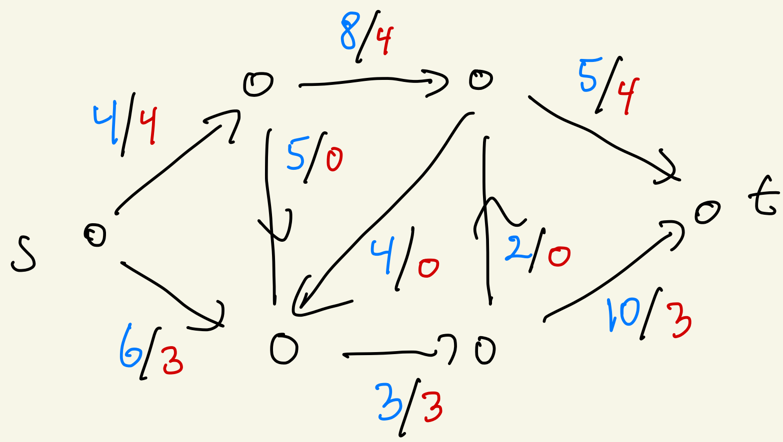
New (s,t)-flow of value 5



flow of value 5

We cannot increase the flow value if we only increase flow in some edges!

But there is a better flow:



flow of value 7

And it is maximum, but how do we prove this?

And how can we find a maximum (s,t)-flow efficiently?

Definition Let $N=(V,E,c)$ be a network with source s and sink t. An (s,t)-cut is a partition $V=S\cup T$ when $T=V\setminus S$ and $s\in S$, $t\in T$. The capacity of the (s,t)-cut (S,T) is

$$c(S,T)=\sum_{\substack{u\in S\\ v\in T}} c(u,v)$$

Lemma Let $N=(V,E,c)$ be a network and f an (s,t)-flow in N. Then for every (s,t)-cut (S,T) in N we have

$$|f|=f(S,T)-f(T,S)$$

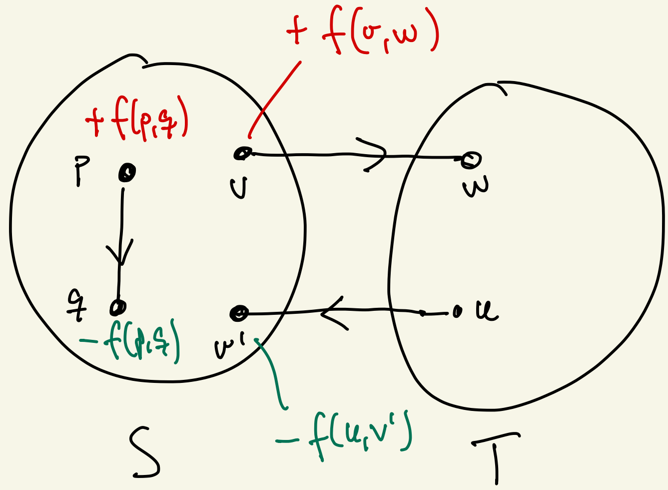
proof

$$|f|=\sum_{v\in S} b_f(v)$$

$$=\sum_{v\in S}\left(\sum_{(v,w)\in E} f(v,w)-\sum_{(u,v)\in E} f(u,v)\right)$$

$$=\sum_{v\in S}\sum_{w\in T} f(v,w)-\sum_{\substack{u\in T\\ v\in S}} f(u,v)=f(S,T)-f(T,S)$$

See next page!



- arcs inside S contribute 0 to sum
- arcs S → T contribute with + to sum
- arcs T → S contribute with − to sum

Lemma For every (s,t)-cut (S,T) in N = (V,E,c) and every (s,t)-flow f in N we have

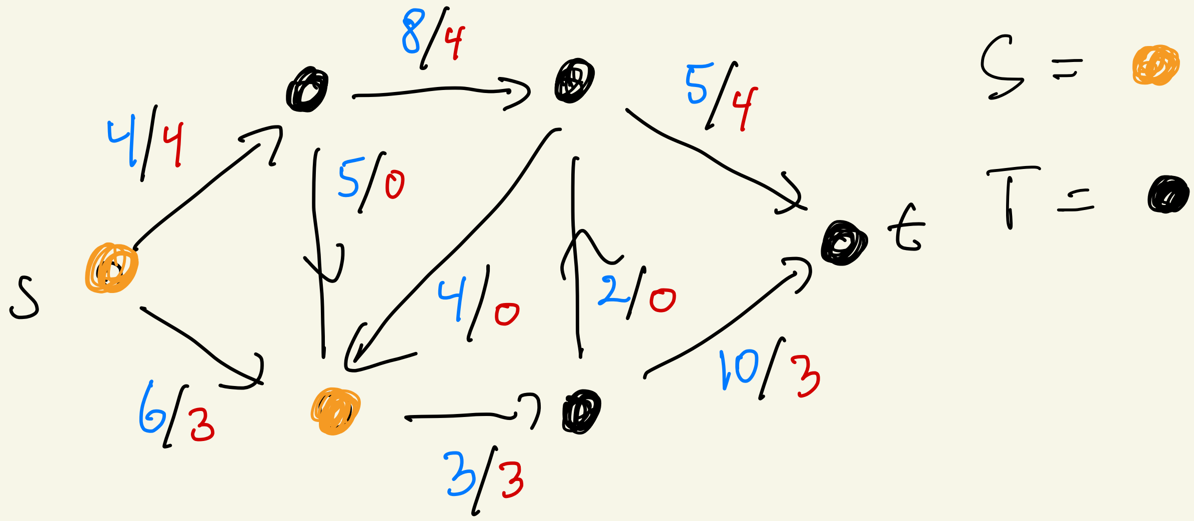
$$|f| \le c(S,T)$$

Proof: from previous Lemma

$$|f| = f(S,T) - f(T,S)$$

$$\le c(S,T) - 0 \qquad \text{as } f(u,v) \le c(u,v) \text{ and } f(u,v) \ge 0$$

$$= c(S,T)$$



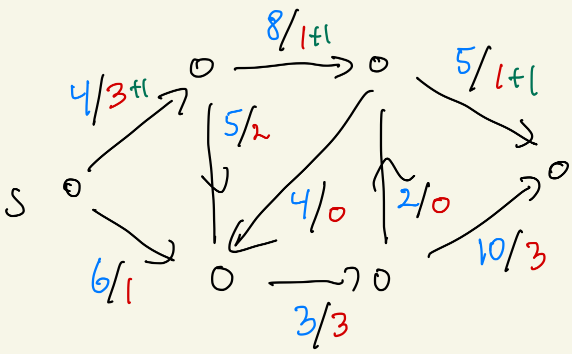
$$|f| = 7 = c(S,T)$$

so $|f|$ is maximum as

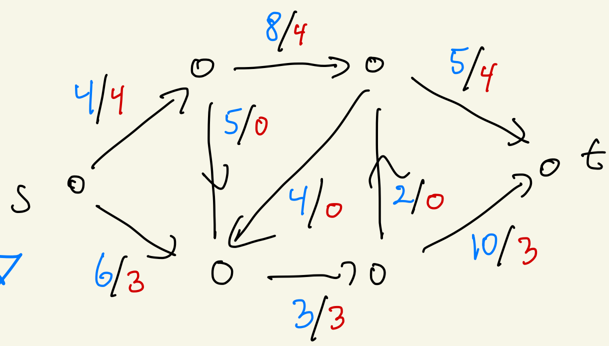
$|f| \le c(S,T)$ for every (s,t)-cut

and we achieved equality!

How can we go from



to



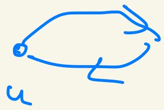
Residual networks

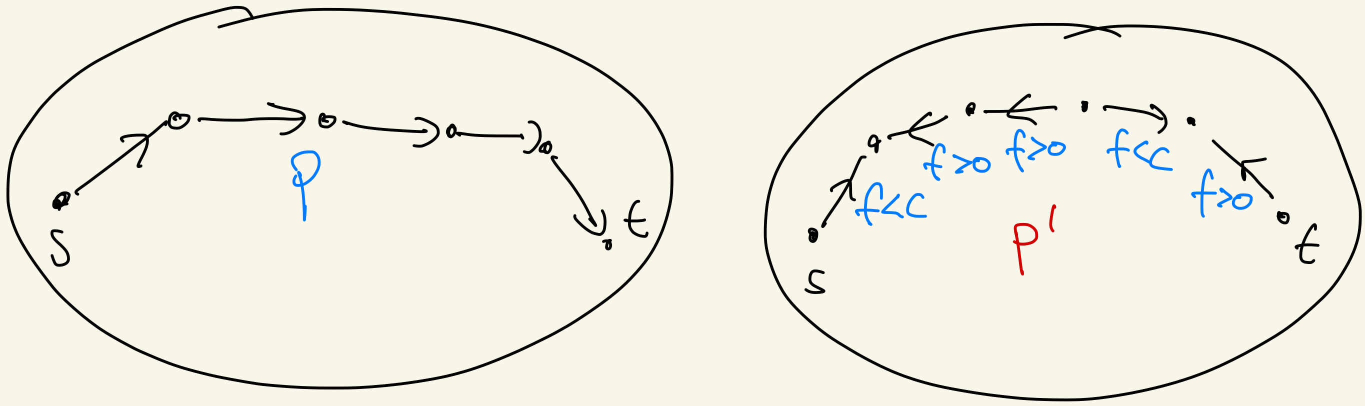
Let $N=(V,E,c)$ and let f be an (s,t)-flow in N

The residual network N_f of N with respect to f is $N_f=(V,E_f,c_f)$ where

$$c_f(u,v)=\begin{cases} c(u,v)-f(u,v) & \text{if } (u,v)\in E\\ f(v,u) & \text{if } (v,u)\in E\\ 0 & \text{otherwise}\end{cases}$$

and $E_f=\{(u,v)\in V\times V \mid c_f(u,v)>0\}$

(recall that we assume no  in N)



N_f

P directed $s\to t$ path in N_f $\Leftrightarrow$ P' oriented path from s to t in N with forward and backwards arcs

Every directed (s,t)-path P in N_f corresponds to a path like this in N:

$$P' = \circ \xrightarrow{f<c} \circ \xrightarrow{f<c} \circ \xleftarrow{f>0} \circ \xleftarrow{f>0} \circ \xrightarrow{f<c} \circ \xleftarrow{f>0} \circ \xrightarrow{f<c} \circ\, t$$

Let $\delta_+(P) = \min\{ c(u,v) - f(u,v) \mid (u,v) \text{ forward arc on } P' \}$

$\delta_-(P) = \min\{ f(u,v) \mid (u,v) \text{ backward arc on } P' \}$

and $\delta(P) = \min\{ \delta_+(P), \delta_-(P) \}$

Then we can increase the flow by $\delta(P)$ units as follows

$$s \circ \xrightarrow{+\delta(P)} \circ \xrightarrow{+\delta(P)} \circ \xleftarrow{-\delta(P)} \circ \xleftarrow{-\delta(P)} \circ \xrightarrow{+\delta(P)} \circ \xleftarrow{-\delta(P)} \circ \xrightarrow{+\delta(P)} \circ\, t$$

corresponds to this flow f_P in N_f:

$$s \circ \xrightarrow{\delta(P)} \circ \xrightarrow{\delta(P)} \circ \xrightarrow{\delta(P)} \circ \xrightarrow{\delta(P)} \circ \xrightarrow{\delta(P)} \circ \xrightarrow{\delta(P)} \circ \xrightarrow{\delta(P)} \circ\, t$$

$$\boxed{\text{Result (denoted } (f \uparrow f_P)) \text{ is an } (s,t)\text{-flow of value } |f| + |f_P| = |f| + \delta(P)}$$

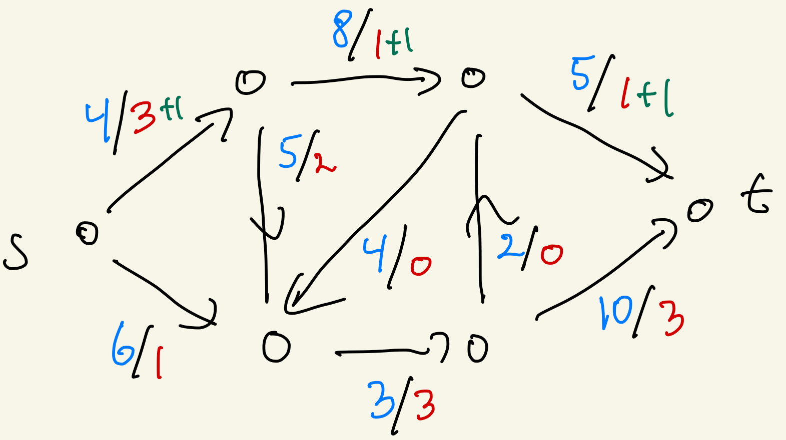
See Lemma 26.1 – Corollary 26.3 in Cormen for calculations but the idea is simple:

- $0 \le (f \uparrow f_p)(u,v) \le c(u,v)$ by definition of $\delta(P)$
- $(f \uparrow f_p)$ is an (s,t)-flow since we add the same amount of flow into each $v \ne s,t$ as out of it
- $|f \uparrow f_p| = |f| + |f_p| = |f| + \delta(P)$ as we increase the flow by $\delta(P)$ on precisely one arc out of s

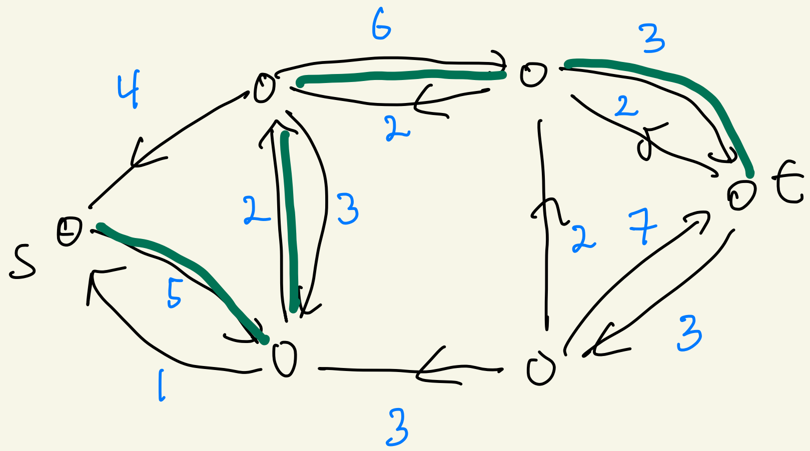
We call a directed (s,t)-path P in N_f an augmenting path and its capacity is the value $\delta(P)$ that we calculated

Back to our example

N, f

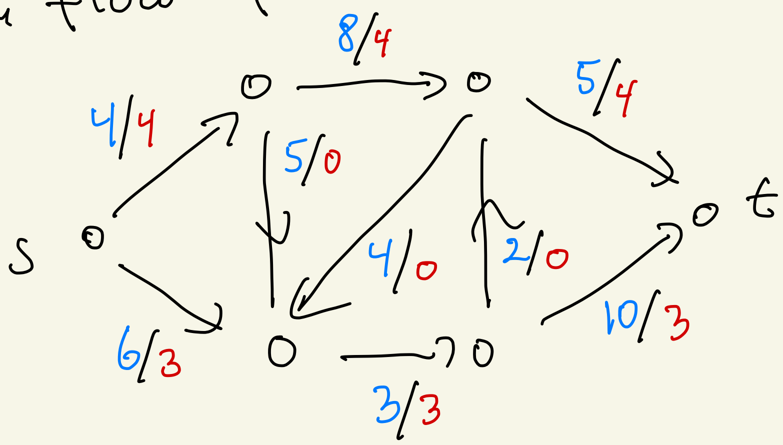


N_f :



c_f in blue

P is the green path in N_f. It has capacity $\delta(P) = \min\{5, 2, 6, 3\} = 2$ and if we let f_P be an (s,t)-flow of value 2 along P then $(f \uparrow f_P)$ is the flow of value 7 we had before:



Ford Folkerson method

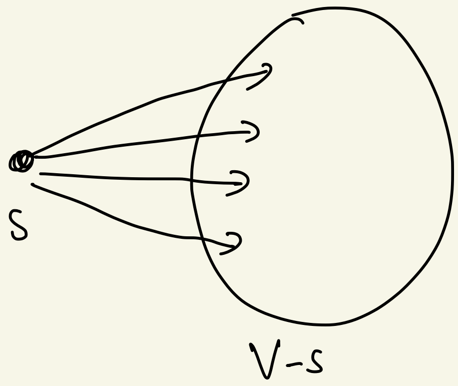
input: a network $N = (V, E, c)$ integer-valued and $s \neq t$ vertices of V

output a maximum (s,t)-flow f in N

1. $f(u,v) := 0 \quad \forall (u,v) \in E$
2. construct N_f
3. while $\exists (s,t)$-path P in N_f do
4. $\quad \delta(P) := \min \{ c_f(u,v) \mid (u,v) \text{ on } P \}$
5. $\quad f_p \leftarrow$ flow of $\delta(P)$ units along P in N_f
6. $\quad f \leftarrow f \uparrow f_p$
7. $\quad$ construct N_f
8. end
9. output f

Note

- As long as N_f contains an augmenting path P we know that f is Not maximum as $|f \uparrow f_p| = |f| + |\delta(P)| > |f|$



- The sum of the capacities of the arcs from s to $V-s$ is a finite integer (as we assumed $c(u,v) \in \mathbb{Z}$)
- The value of f increases by at least one in each iteration (augmenting path

so the algorithm will stop.

How do we prove that f IS maximum when the algorithm stops?

Max-flow min-cut theorem

(1) f is a maximum valued (s,t)-flow in $N=(V,E,C)$

$\Updownarrow$ (2) there is no (s,t)-path in N_f

$\Updownarrow$ (3) $|f| = c(S,T)$ for some (s,t)-cut (S,T)

Proof

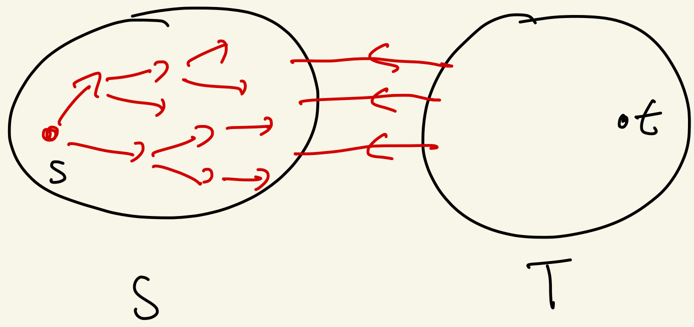
(1) $\Rightarrow$ (2) if P is an (s,t)-path in N_f then $\delta(P) > 0$

so $|f \uparrow f_p| = |f| + \delta(P) > |f| \rightarrow \Leftarrow$

so no (s,t)-path in N_f

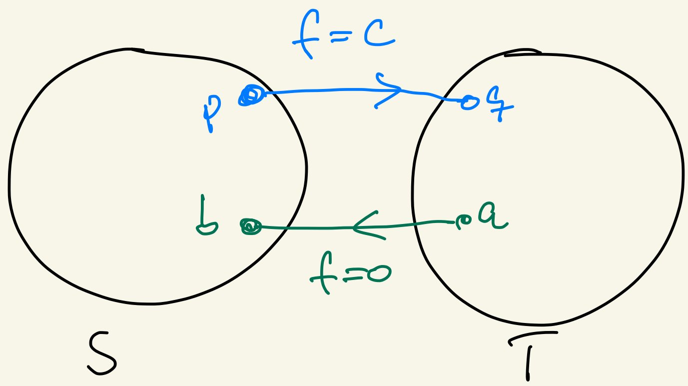
(3) $\Rightarrow$ (1) if $|f| = c(S,T)$ then f is maximum as $|f| \leq c(S',T')$ for every (s,t)-cut (S',T')

(2) ⇒ (3): Suppose there is no (s,t)-path in N_f



S is the set of vertices reachable from s in N_f
t is not reachable so $t \in T$

in N_f there is no edge (u,v) with $u \in S$, $v \in T$
This means that in N we have



as $(p,q) \in E_f$ if $f(p,q) < c(p,q)$

as $(b,a) \in E_f$ if $f(a,b) > 0$

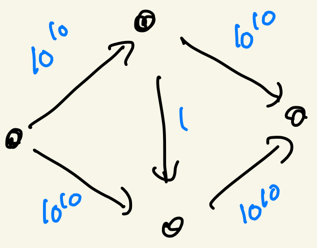
Now we have

$$
\begin{aligned}
|f| &= f(S,T) - f(T,S) \\
&= c(S,T) - 0 \\
&= c(S,T)
\end{aligned}
$$

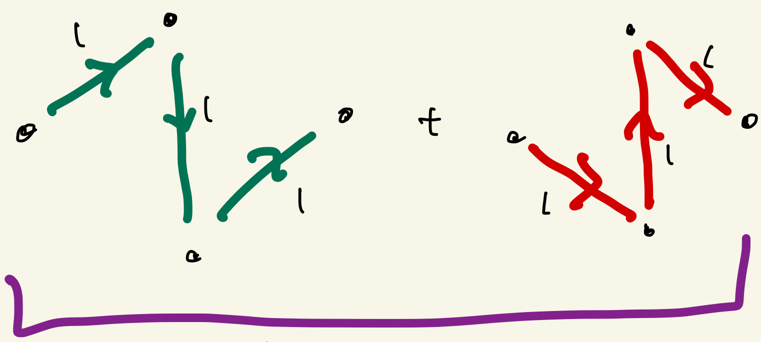
So (3) holds $\square$.

Notes of Ford-Fulkerson's algorithm

- We are just looking for an (s,t)-path to augment along so the algorithm may make very small progress



N



repeat 10^{10} times

running time $O(|f^*||E|)$ f^* maximum flow

so not polynomial

- If capacities may be irrational numbers then the algorithm may never terminate!

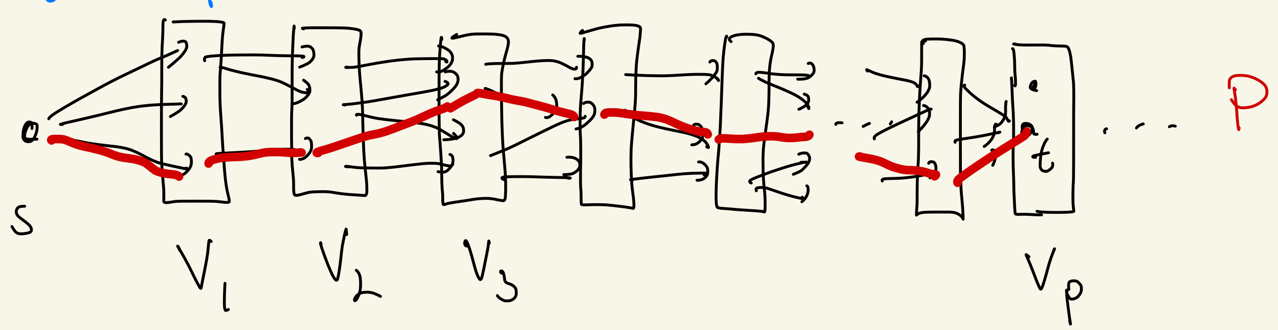
Can we get a better algorithm if we choose the augmenting paths more cleverly?

Edmonds - Karp - Algorithm

As the Ford-Fulkerson algorithm but paths in N_f are shortest (s,t)-paths.

This leads to polynomial running time $O(|V||E|^2)$

Recall Breath-First-search distance froms in N_f:

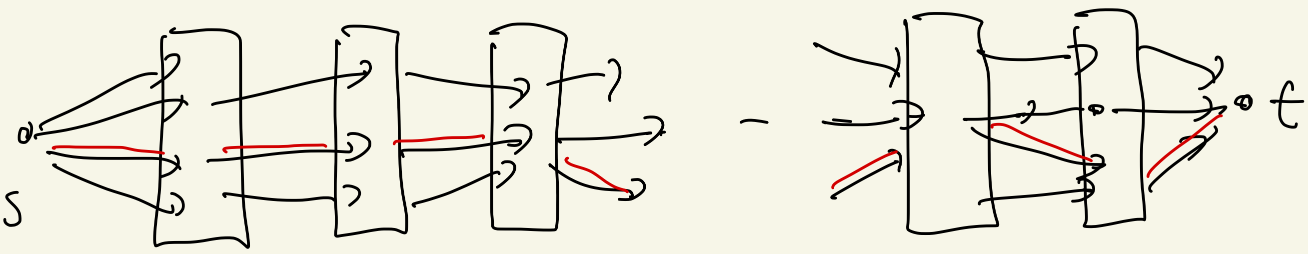


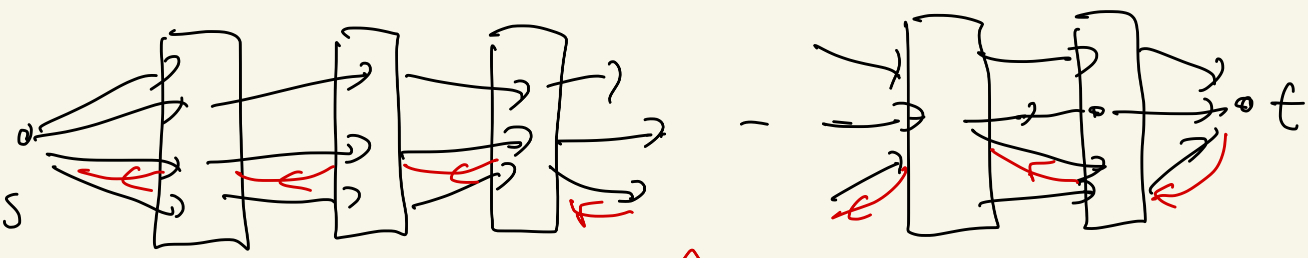
Let p be current distance from s to t in the residual network N_f

Which new edges will appear in the residual network of $f \uparrow f_P$?

Only arcs opposite to arcs on P !

flow only changes on arcs corresponding to pairs joined by an edge on P





possible new edges in $N_{f \uparrow f_P}$

Consequence of this:

The distance from s to t in $N_{f \uparrow f_P}$ is at least as large as the distance from s to t in N_f

So the distance dist(s,t) from s to t in current residual network is an increasing functi

- $\mathrm{dist}(s,t) \geq 1$ in the beginning
- $\mathrm{dist}(s,t) \leq n-1$ as long as there is an (s,t)-path
- at most $|E|$ paths with same distance $\mathrm{dist}(s,t) = p$

  during the algorithm: at least one edge of P is not in $N_{f \uparrow f_P}$ by definition of $\delta(P)$
- Thus at most $(|V|-1) \cdot |E| = O(|V||E|)$ augmenting paths
- Each augmenting path is found using BFS in time $O(|V|+|E|)$
- In total the algorithm spends $O(|V||E| \cdot (|V|+|E|)) = O(|V||E|^2)$ steps to find a maximum flow

<u>Note</u> the algorithm works for <u>all</u> capacity functions (not just integer functions)