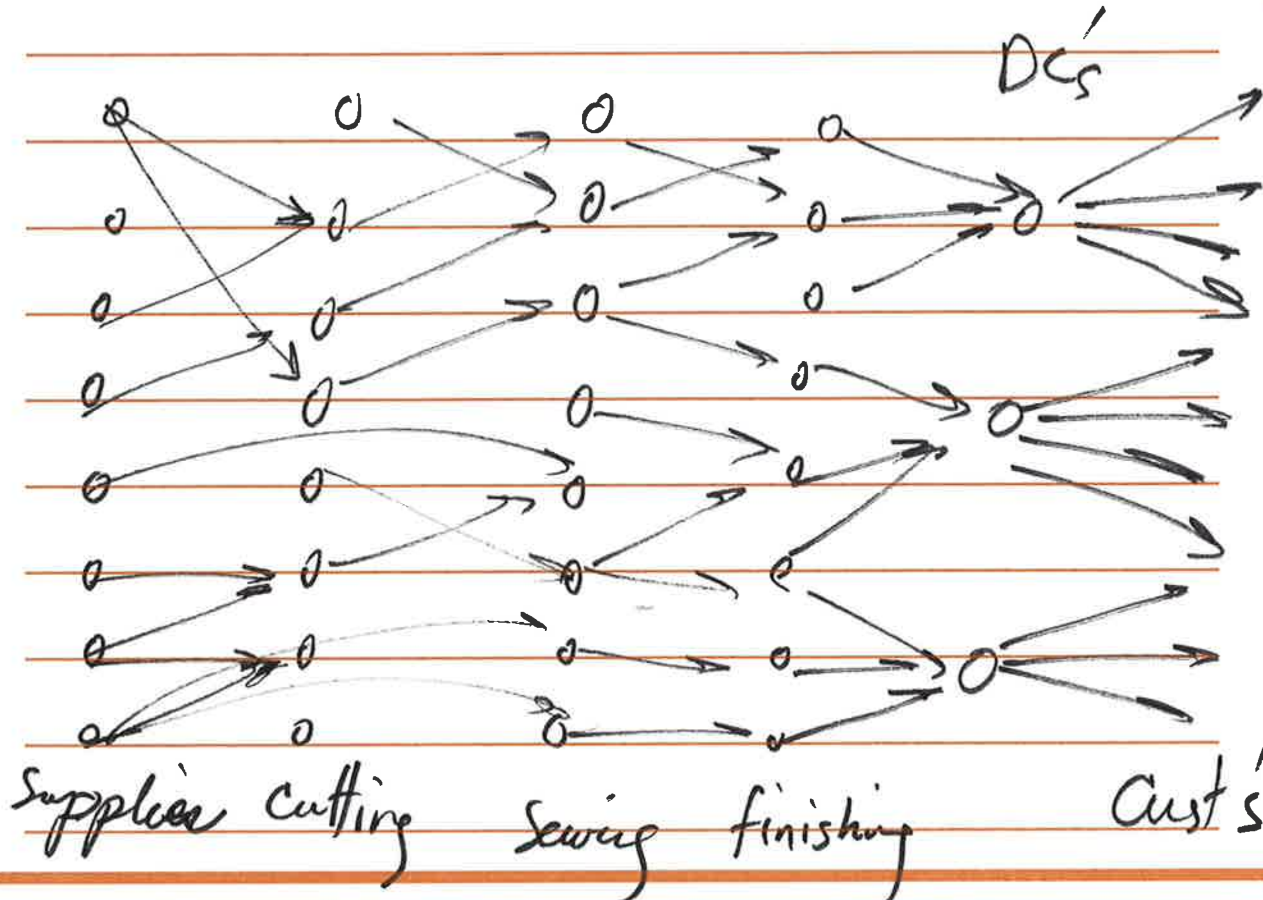
Network Flow




Def. A <u>flow Network</u> is a directed graph

$G = (V, E)$ with following features:

- each edge $e$ has a non-negative capacity $c_e$.
- Has a single source node $s \in V$
- " " " sink " $t \in V$
- 3 assumptions:
  - no edges enter $s$ & leave $t$
  - at least one edge is connected to each node.

- all capacities are integers

Notation: we call $f(e)$ flow over edge $e$.

$f(e)$ has following properties:

1- For each $e \in E$ $\quad 0 \leq f(e) \leq c_e$

(capacity condition)

$\sum$

2- For each node $v$ other than $s$ & $t$

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$

(Conservation of flow)

Assumption: flow is <u>steady state</u>

<u>Def.</u> For a steady state flow, the value of a flow is

$$v(f) = \sum_{e \text{ out of } s} f(e)$$

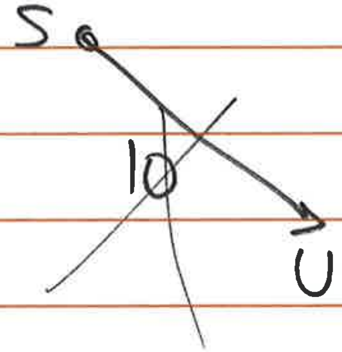
Can also say $f^{out}(s) = \sum_{e \text{ out of } s} f(e)$

~~Can also~~

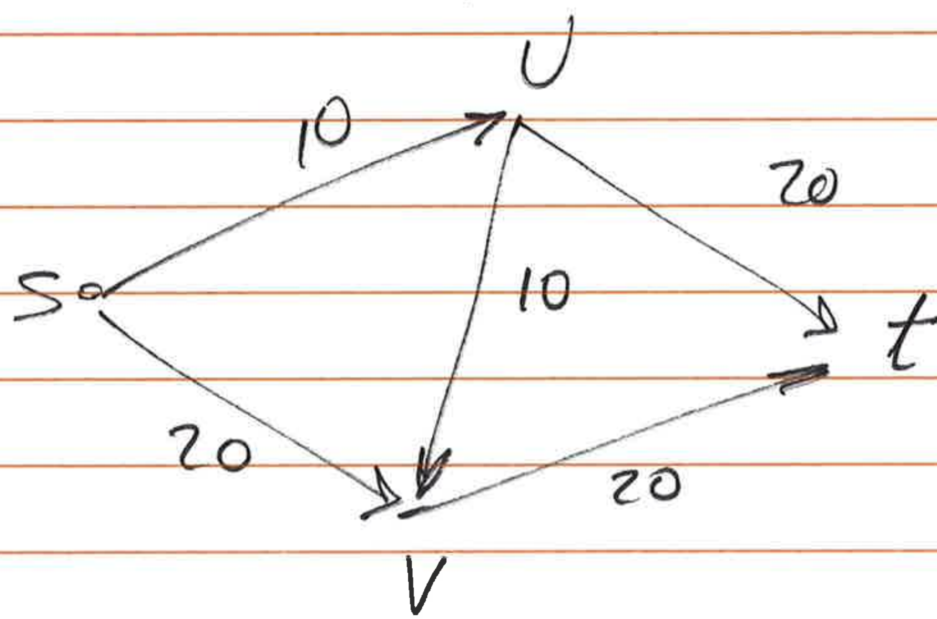
$$f^{out}(v) = \sum_{e \text{ out of } v} f(e)$$

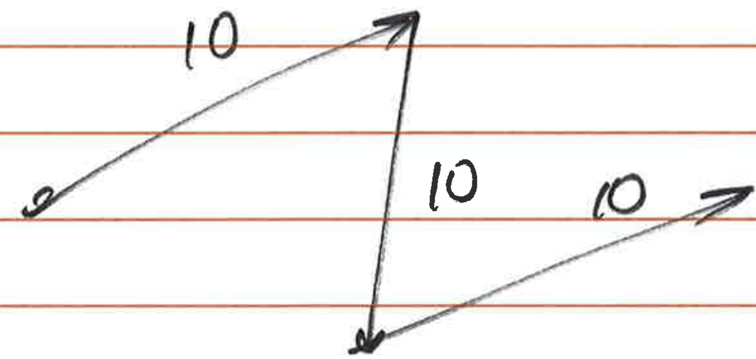
$$f^{in}(v) = \sum_{e \text{ into } v} f(e)$$

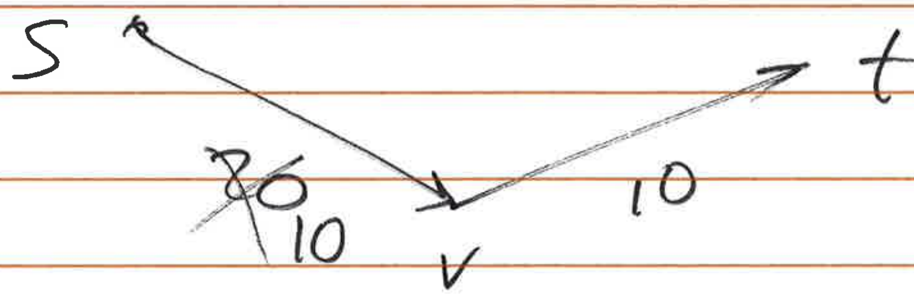
Conservation of flows: $f^{in}(v) = f^{out}(v)$




try #2 lowest cap. edges first






<u>Def.</u> $G_f$ is the residual graph of $G$ with the following definition:

- $G_f$ has the same set of nodes as $G$
- for each edge $\underline{e}$ w/ $f(e) < c_e$ we include $e$ in $G_f$ with cap $c_e - f(e)$
- for each edge $e$ w/ $f(e) > 0$ we include edge $e'$ in opposite direction to $\underline{e}$ in $G_f$ w/ cap $f(e)$

$G \qquad v \xrightarrow[f(e)=c_e]{e} u$

$G_f \qquad v \xleftarrow[f(e)]{e'} u$

---

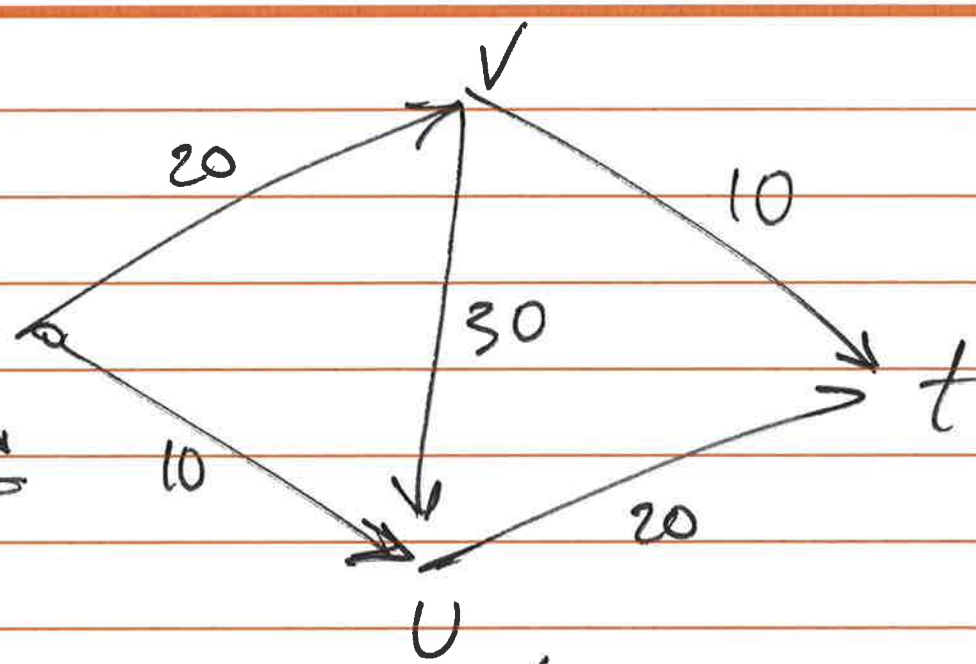
$G \qquad v \xrightarrow[f(e)=0]{e} u$

$G_f \qquad v \xrightarrow[c_e]{} u$

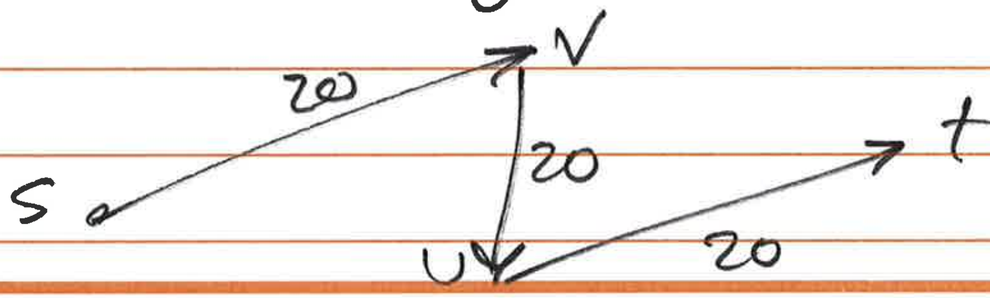
$G \qquad v \xrightarrow[0<f(e)<c_e]{e} u$

$$G_f \qquad \begin{tikzcd} v \arrow[r, bend left, "c_e - f(e)"] & u \arrow[l, bend left, "f(e)"] \end{tikzcd}$$

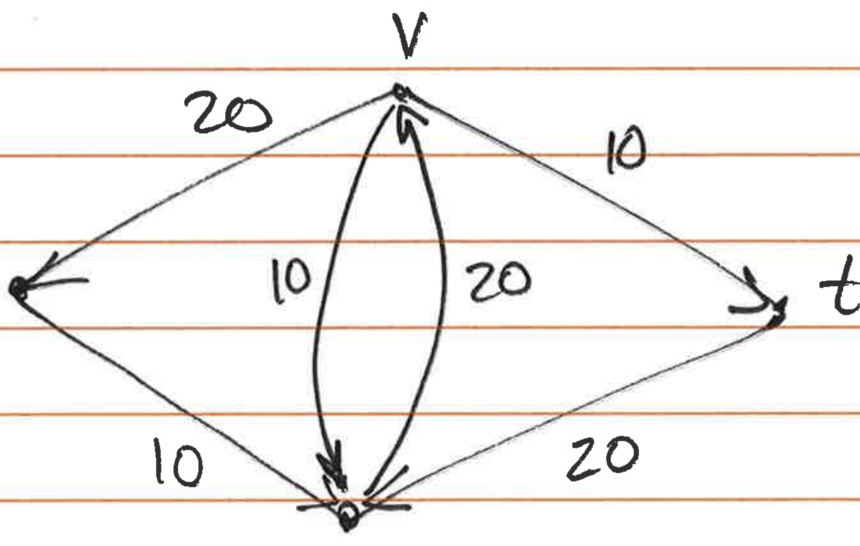
$G$



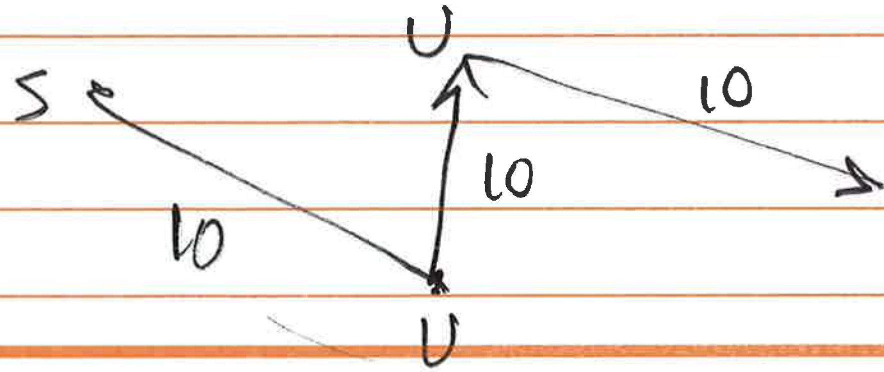

$f_1$



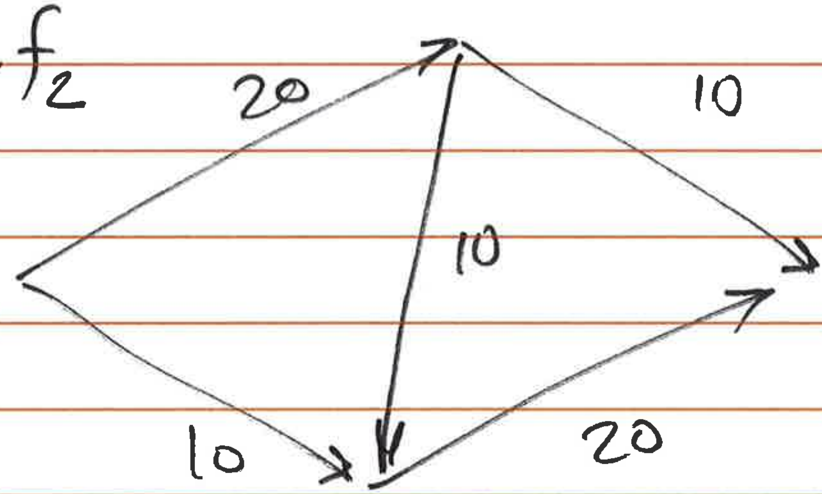

$G_{f_1}$




$f_2$



$f = f_1 + f_2$




Def. If $P$ is a simple path from $s$ to $t$ in $G_f$, then bottleneck $(p)$ is the min. residual cap. of any edge on $P$.

General approach

- Find a path from $s$ to $t$
- Find the bottleneck
- Push flow on that path equal to bottleneck value
- repeat.

Augment( f, c, P )
let $\underline{b}$ = bottleneck (P)
for each edge (v, u) ∈ P
if e = ( v, u) is a forward edge
then ~~f(e) in~~ increase f(e)
in G by $\underline{b}$.
else find edge e = (u, v)
decrease f(e) in G by $\underline{b}$
endif
endfor

Return (f)

Need to prove that the result of augment ( ) is a new flow f'
i.e. f' must be <u>valid</u>.

1- Cap. Condition ✓

Show for each edge $e \in E$

$$0 \le f'(e) \le c_e$$

A- if $e = (u, v)$ is a forward edge

$$\underbrace{f(e) + \text{bottleneck}(p)}_{} \le c_e - \cancel{f(e)} + \cancel{f(e)}$$

$$0 \le f'(e) \le c_e$$

B- if $e = (u, v)$ is a backward edge then

$$\text{bottleneck}(p) \le f(e)$$

$$\underbrace{f(e) - \text{bottleneck}(p)}_{} \ge \cancel{f(e)} - \cancel{f(e)}$$

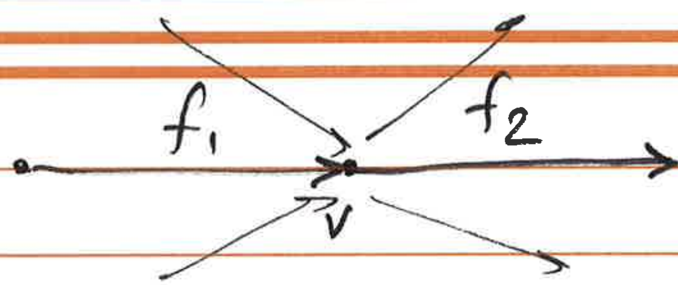
$$c_e \ge f'(e) \ge 0$$

2- We need to prove that Conservation of flow is maintained

look at node v

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$

before




after $\bullet \xrightarrow{f_1+b} \xrightarrow{f_2+b}$

Max flow ( G, s, t, c )

Initially $f(e) = 0$ for all e in E

while there is an s-t path in residual graph $G_f$

O(m) let P be a simple s-t path in $G_f$

O(n) $f' = $ augment ( f, c, P )

$f = f'$

update $G_f$

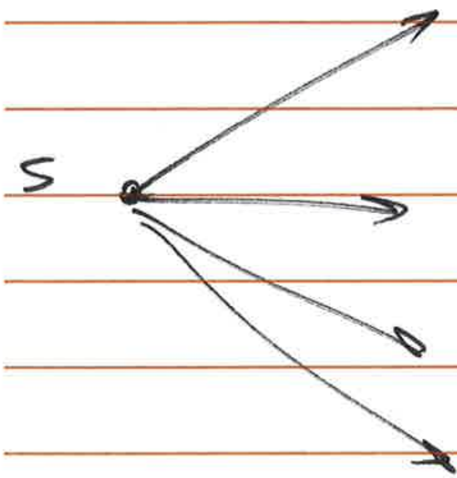
end while

Ford-Fulkerson Algorithm

① while loop ends

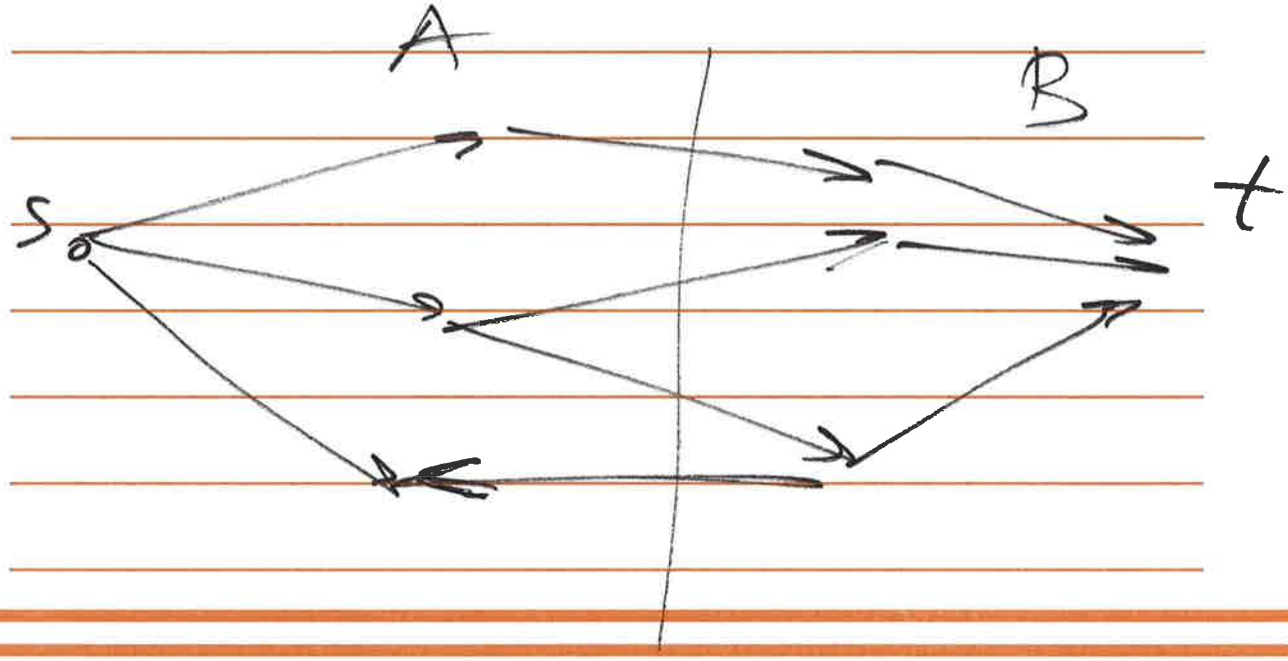
FACT: At every intermediate step, flow values {f(e)} and residual cap's in $G_f$ are integers.

take O(Cm)




$$v(f) = \sum_{e \text{ out of } s} f(e) \leq \sum_{e \text{ out of } s} c_e = \underline{C}$$

A cut divides nodes in a flow network into 2 sets A & B such that $s \in A$ & $t \in B$




Notation:

$C(A,B)$ = Cap. of a cut $(A,B)$

$$C(A,B) = \sum_{e \text{ out of } A} c_e$$

FACT: Let $f$ be any s-t flow, and $(A,B)$ any s-t cut

then $v(f) = f^{out}(A) - f^{in}(A)$

## useful Consequence

Max. value of the flow $\leq$ Cap. of the (A,B) cut

Proof:

$$v(f) = f^{out}(A) - f^{in}(A)$$

$$v(f) \leq f^{out}(A)$$

$$\leq \sum_{e \text{ out of } A} f(e) \leq \sum_{e \text{ out of } A} c_e$$

$$v(f) \leq \sum_{e \text{ out of } A} c_e$$

$$v(f) \leq c(A,B)$$

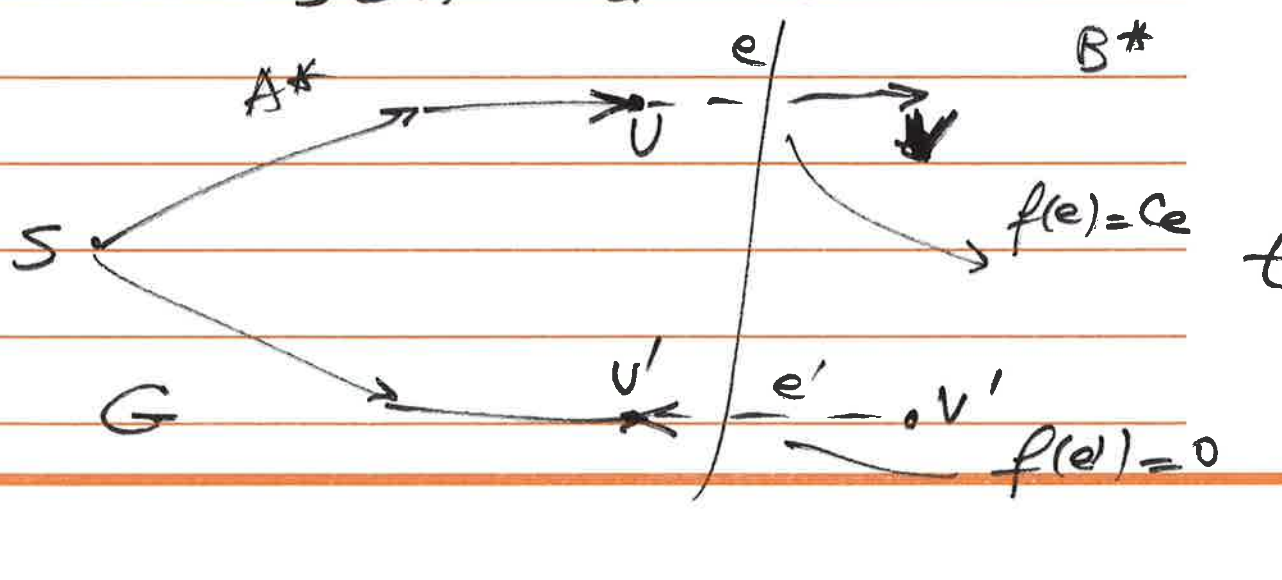
Ford-Fulkerson terminates when the flow $f$ has no s-t path in $G_f$.

claim: If there is no s-t path in $G_f$ then there is an s-t cut $(A^*, B^*)$ where $v(f) = c(A^*, B^*)$

Create sets $A^*$ & $B^*$ such that $A^*$ includes all nodes $v$ where there is an s-v path in $G_f$

$B^* = V - A^*$

$s \in A^*$ & $t \in B^*$



$$v(f) = f^{out}(A^*) - f^{in}(A^*)$$

$$= \sum_{e \text{ out of } A^*} f(e) - \sum_{e \text{ into } A^*} f(e)$$

$$= \sum_{e \text{ out of } A^*} c_e - 0$$

$$v(f) = C(A^*, B^*)$$