24 February 2025

Max Flow Algorithms: Ford - Fulkerson.

Plan

* Ideas for Flow Algorithms
* Announcements
* Ford - Fulkerson

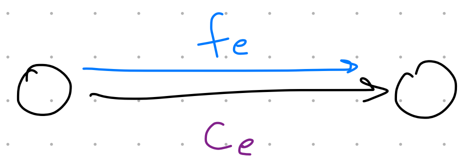
Network Flow Problem

* Given Flow Network $G, s, t, c: E \to \mathbb{R}^+$
* Find Flow $f: E \to \mathbb{R}^+$ subject to

Capacity Constraints

$\forall e \in E$

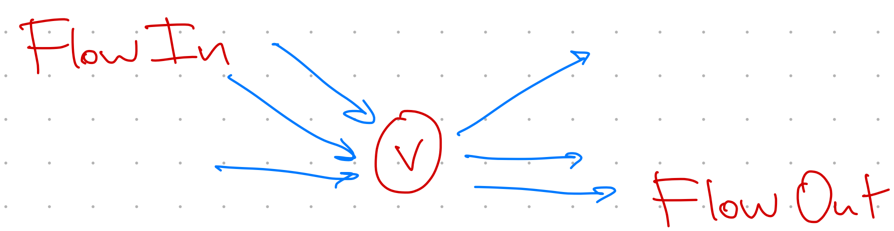
$$0 \leq f_e \leq c_e$$



Conservation Constraints

$\forall v \in V \setminus \{s, t\}$

$$\sum_{uv \in E} f_{uv} = \sum_{vw \in E} f_{vw}$$

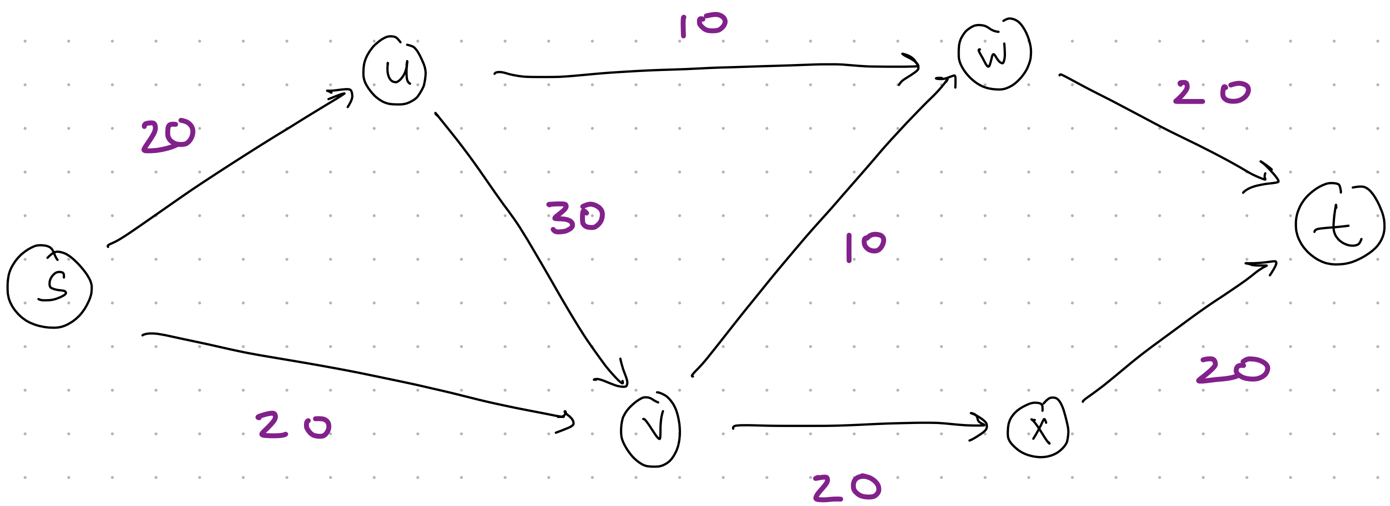


Max Flow. Find Flow f^* that maximizes FlowOut(s)

Attempt #0. Greedy

* Find st-path w/ positive capacity
* Push flow along path

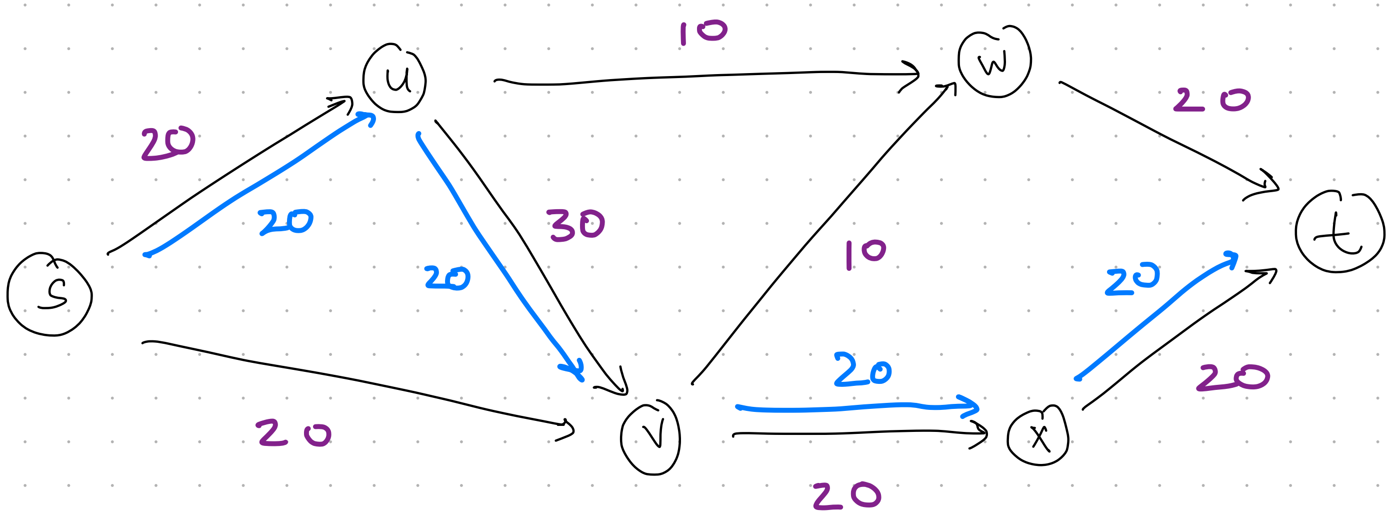
Repeat until No s⇝t paths.



Attempt #0. Greedy

* Find st-path w/ positive capacity
* Push flow along path

Repeat until No $s \rightsquigarrow t$ paths.

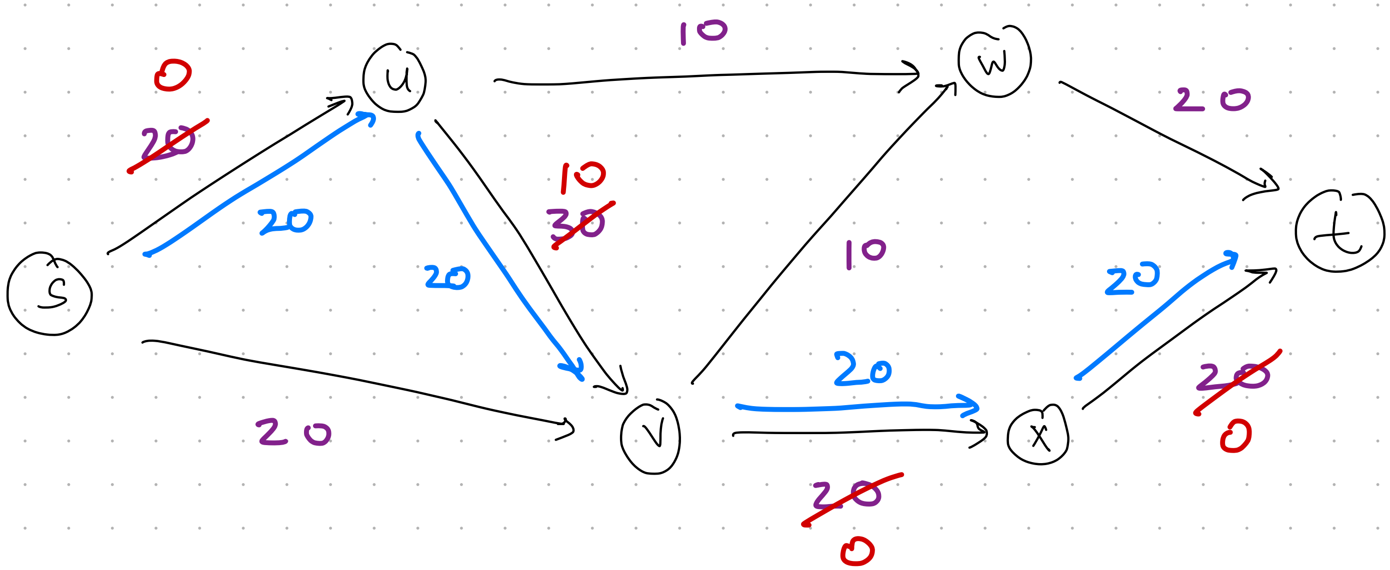


20 $\quad s \rightarrow u \rightarrow v \rightarrow x \rightarrow t$

Attempt #0. Greedy

* Find st-path w/ positive capacity
* Push flow along path

Repeat until No $s \leadsto t$ paths.

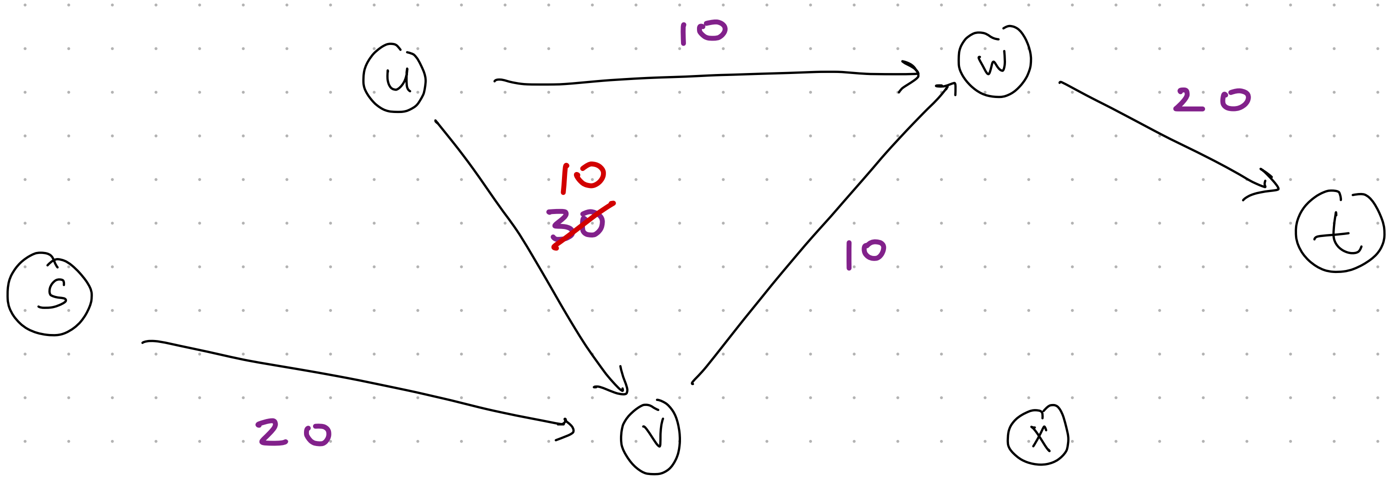


20 $s \to u \to v \to x \to t$

Attempt #0. Greedy

* Find st-path w/ positive capacity
* Push flow along path

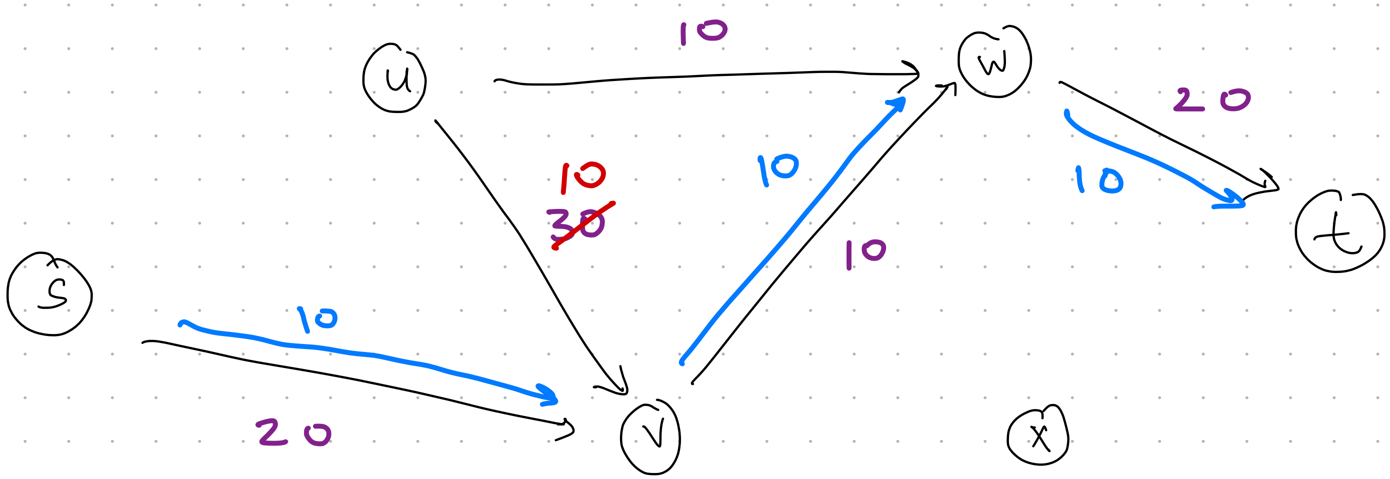
Repeat until No $s \rightsquigarrow t$ paths.



Attempt #0. Greedy

* Find st-path w/ positive capacity
* Push flow along path

Repeat until No $s \rightsquigarrow t$ paths.



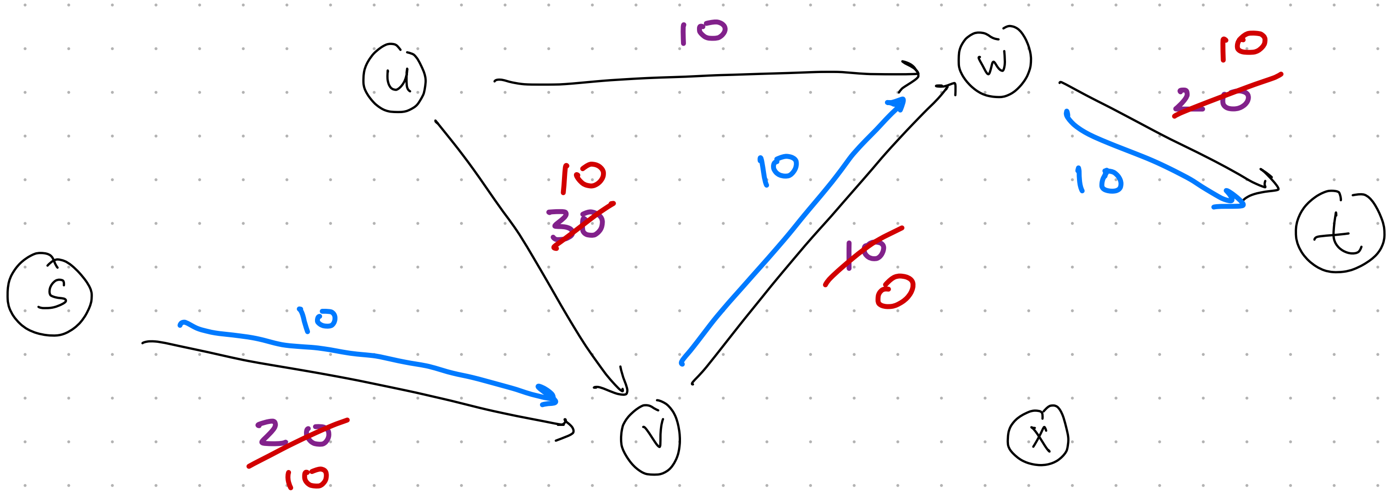
20 $\quad s \to u \to v \to x \to t$

10 $\quad s \to v \to w \to t$

Attempt #0. Greedy

* Find st-path w/ positive capacity
* Push flow along path

Repeat until No $s \rightsquigarrow t$ paths.



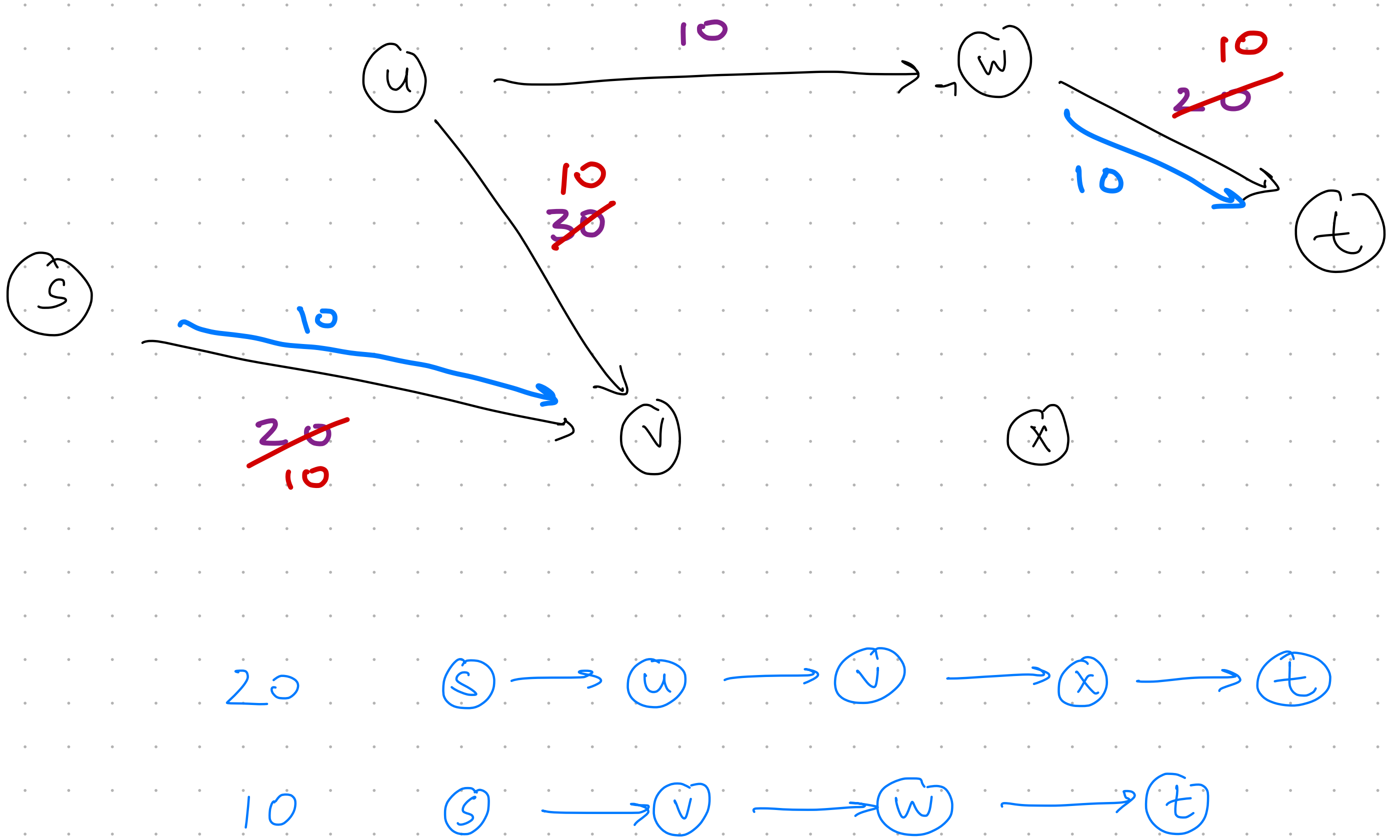
20 $s \to u \to v \to x \to t$

10 $s \to v \to w \to t$

Attempt #0. Greedy

* Find st-path w/ positive capacity
* Push flow along path

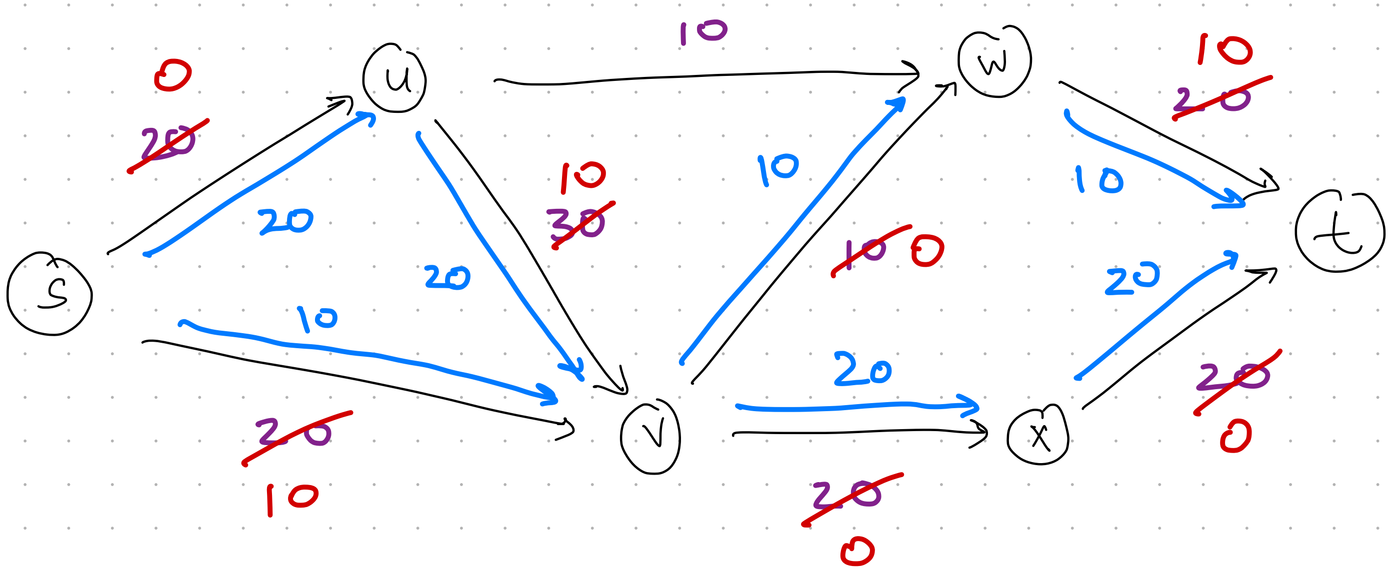
Repeat until No s⇝t paths.



Attempt #0. Greedy

* Find st-path w/ positive capacity
* Push flow along path

Repeat until No s⇝t paths.



20 $s \to u \to v \to x \to t$

10 $s \to v \to w \to t$

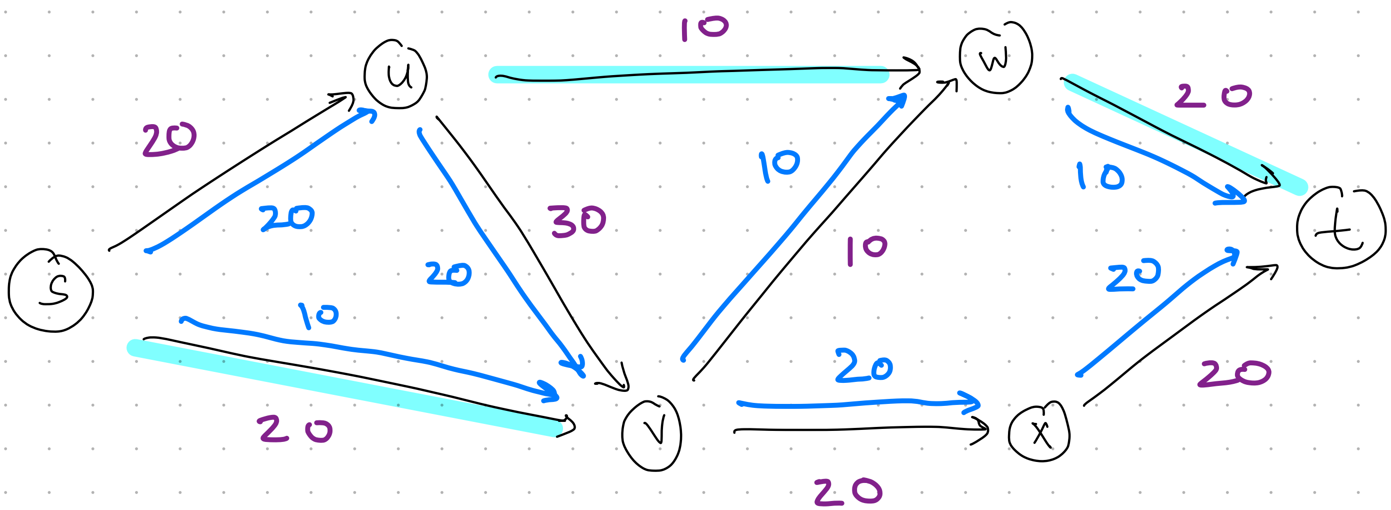
No more $s \to t$ paths

GREEDY FLOW : 30.

Problem w/ Greedy

Myopic

Irrevocable

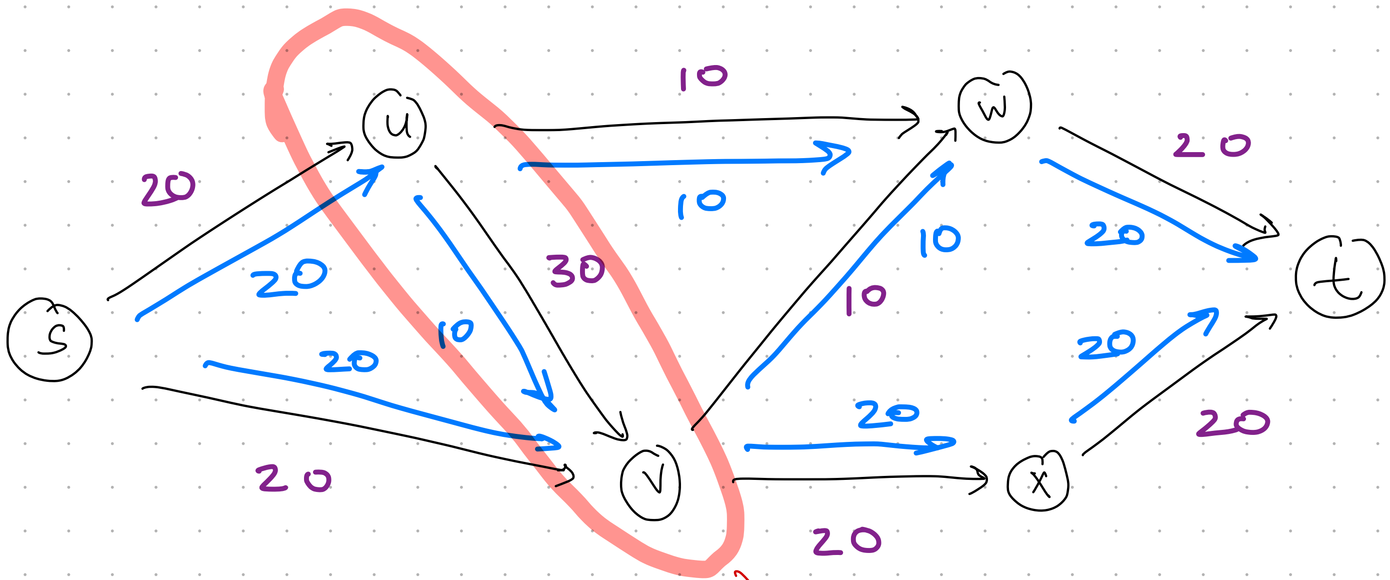


MaxFlow(G) = 40.

Problem w/ Greedy

Myopic

Irrevocable



MaxFlow(G) = 40.

Greedy solution assigned $f_{uv} = 20$.

but $f^*_{uv} = 10$.

Announcements

* Prelim #1 Grades Released After Lecture.

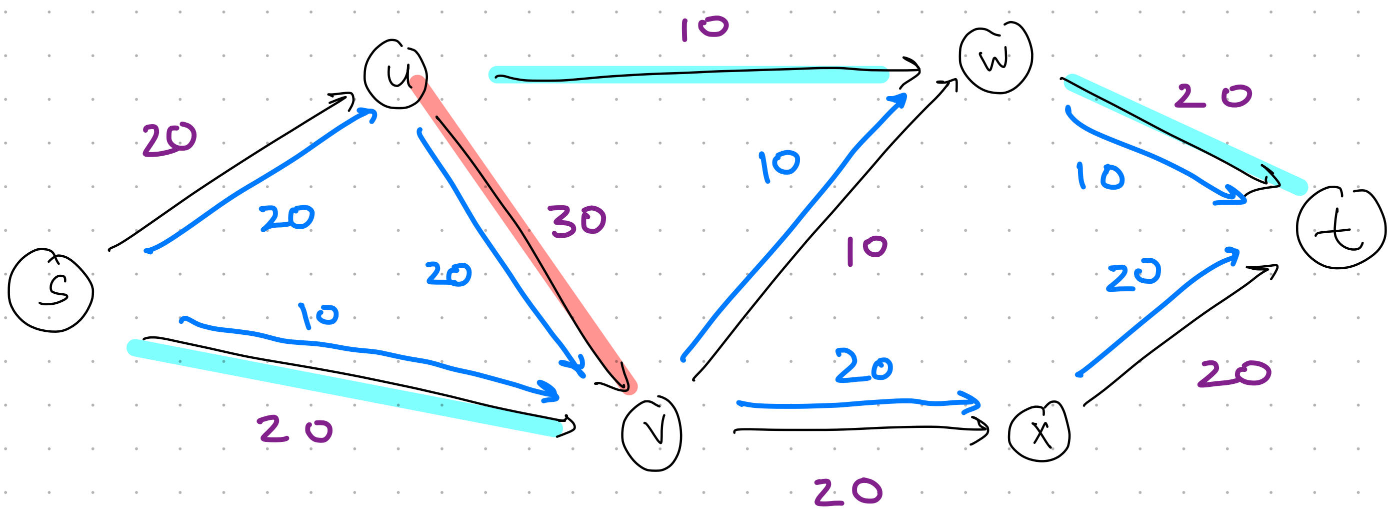
* HW #3 Released Wednesday.

* Recitation Starts again on Saturday.

Problem w/ Greedy

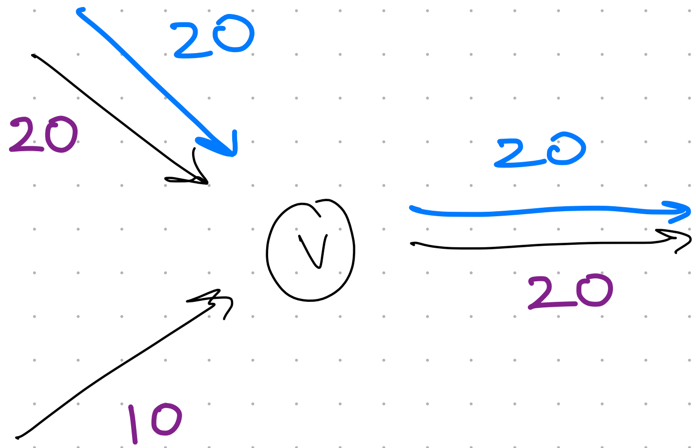
Myopic

Irrevocable



Goal. Systematic way to "Revoke" on flow so far.

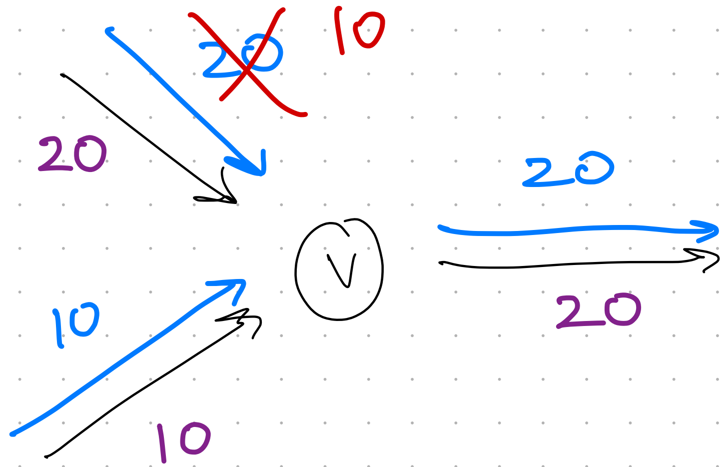
Augmenting a flow



To conserve Flow at v.

$\Delta \text{FlowIn}(v) = \Delta \text{FlowOut}(v)$

Augmenting a flow



To conserve Flow at v.

$$\Delta \text{FlowIn}(v) = \Delta \text{FlowOut}(v)$$

Given a Flow Network G, source s, sink t
w/ capacities $c: E \to \mathbb{R}^+$
and flow $f: E \to \mathbb{R}^+$

The Residual Network

G^f is a flow network w/ same vertex set as G.

and residual capacities $c^f: E \to \mathbb{R}^+$, where

$$c^f(uv) = \begin{cases} \quad \end{cases}$$

Allow us to reason about flow we can still push/unpush

u v

Given a Flow Network G, source s, sink t
w/ capacities $c: E \to \mathbb{R}^+$
and flow $f: E \to \mathbb{R}^+$

<u>The Residual Network</u> G^f is a flow network w/ same vertex set as G.

and residual capacities $c^f: E \to \mathbb{R}^+$, where

$$c^f(uv) = \begin{cases} \emptyset & \text{if } f_{uv} = c_{uv} \\ & \end{cases}$$

NO RESIDUAL

u X v

uv at capacity

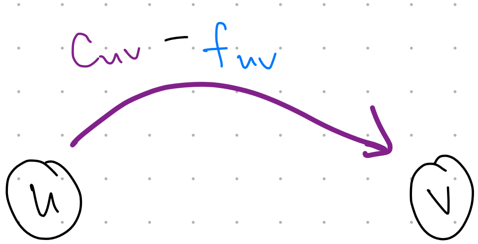
Given a Flow Network G, source s, sink t
w/ capacities $c: E \to \mathbb{R}^+$
and flow $f: E \to \mathbb{R}^+$

The Residual Network G^f is a flow network w/ same vertex set as G.

and residual capacities $c^f: E \to \mathbb{R}^+$, where

$$c^f(uv) = \begin{cases} \emptyset & \text{if } f_{uv} = c_{uv} \\ c_{uv} - f_{uv} & \text{if } f_{uv} < c_{uv} \end{cases}$$

FORWARD RESIDUAL



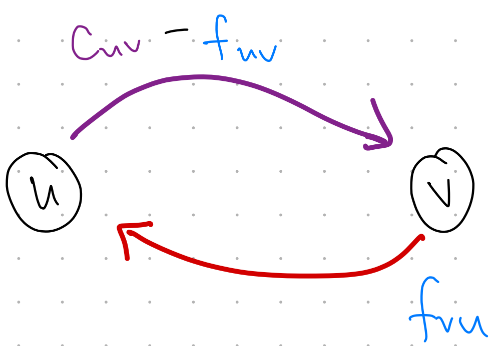
uv still has capacity to push flow

Given a Flow Network G, source s, sink t
w/ capacities $c: E \to \mathbb{R}^+$
and flow $f: E \to \mathbb{R}^+$

The Residual Network G^f is a flow network
w/ same vertex set as G.

and residual capacities $c^f: E \to \mathbb{R}^+$, where

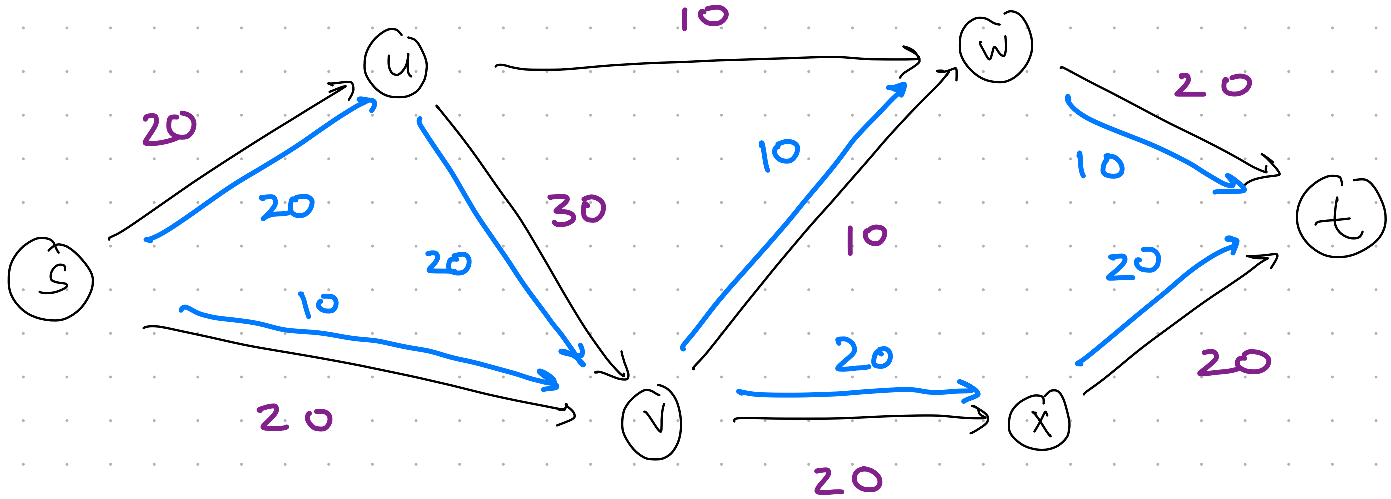
$$c^f(uv) = \begin{cases} 0 & \text{if } f_{uv} = c_{uv} \\ c_{uv} - f_{uv} & \text{if } f_{uv} < c_{uv} \\ f_{vu} & \text{if } f_{vu} > 0 \end{cases}$$



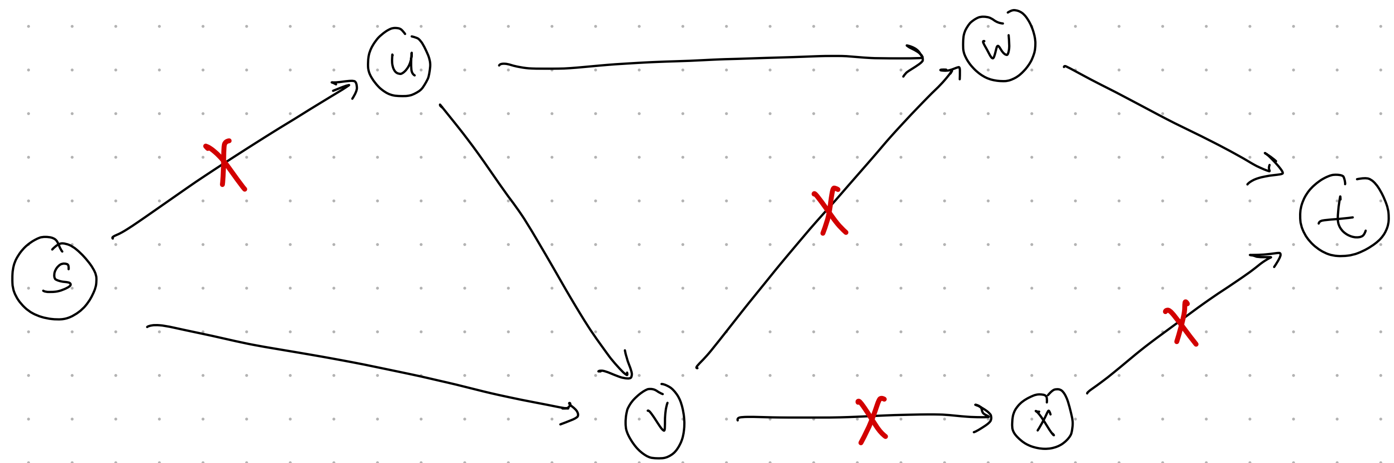
BACKWARD RESIDUAL

vu has capacity to be "unpushed"

Output of Greedy

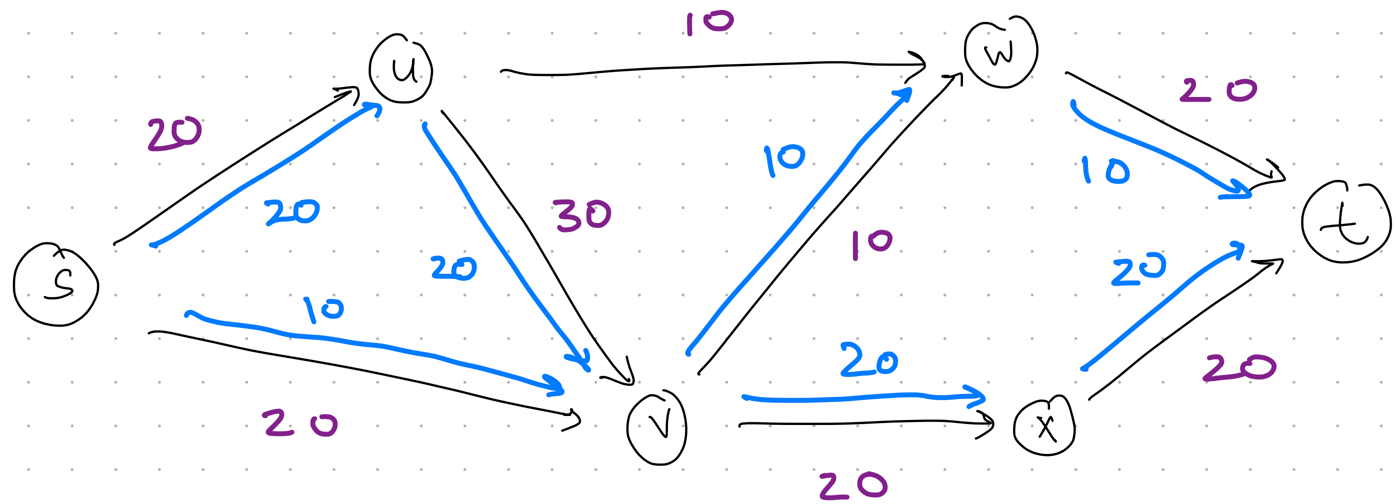


Residual Network

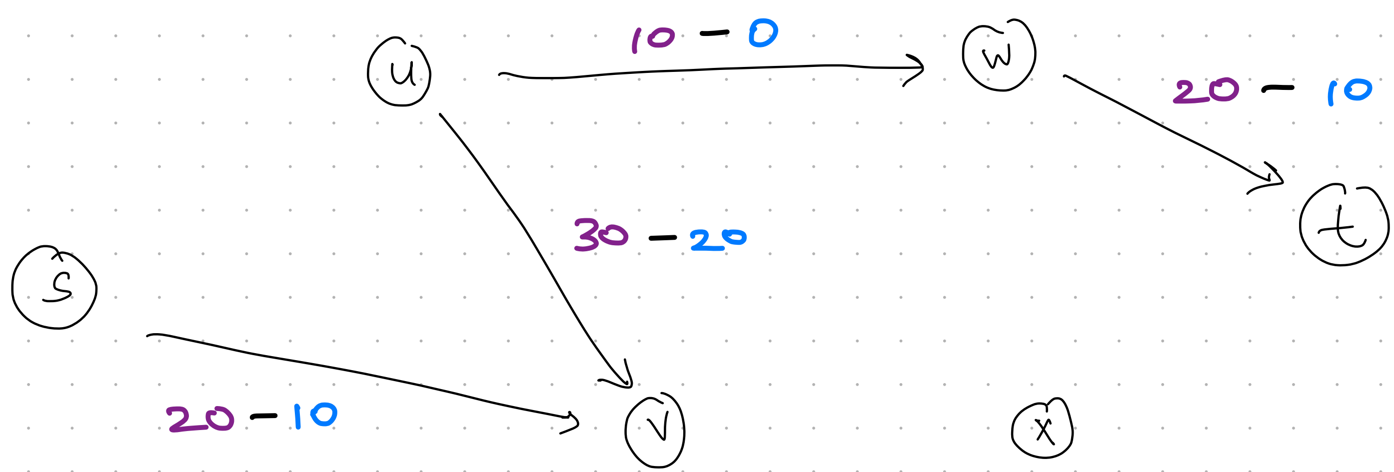


NO RESIDUAL $f_e = c_e$

Output of Greedy

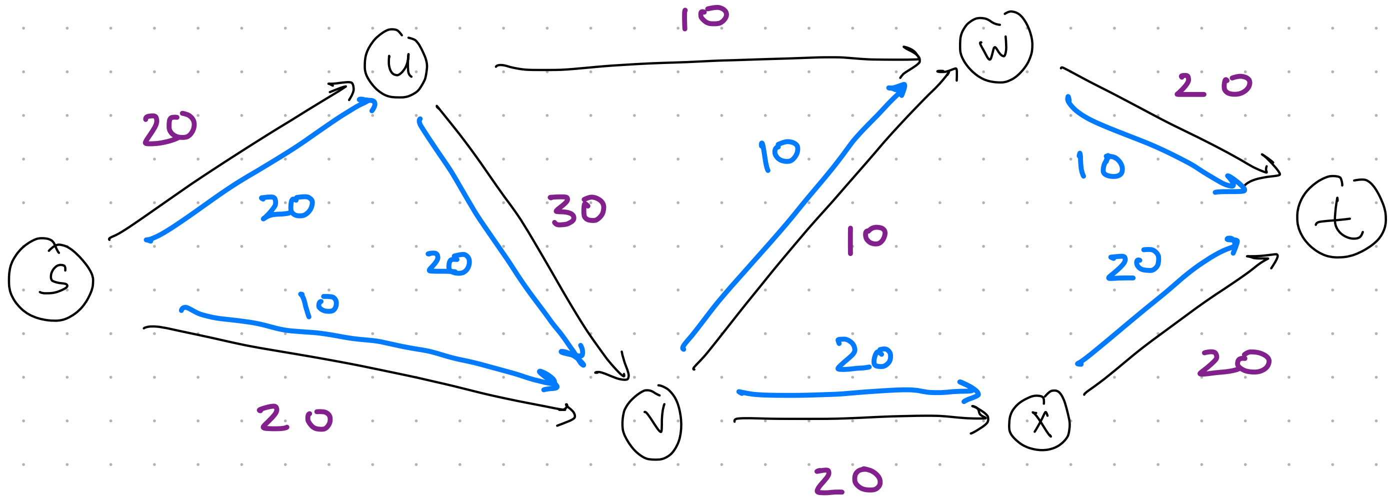


Residual Network

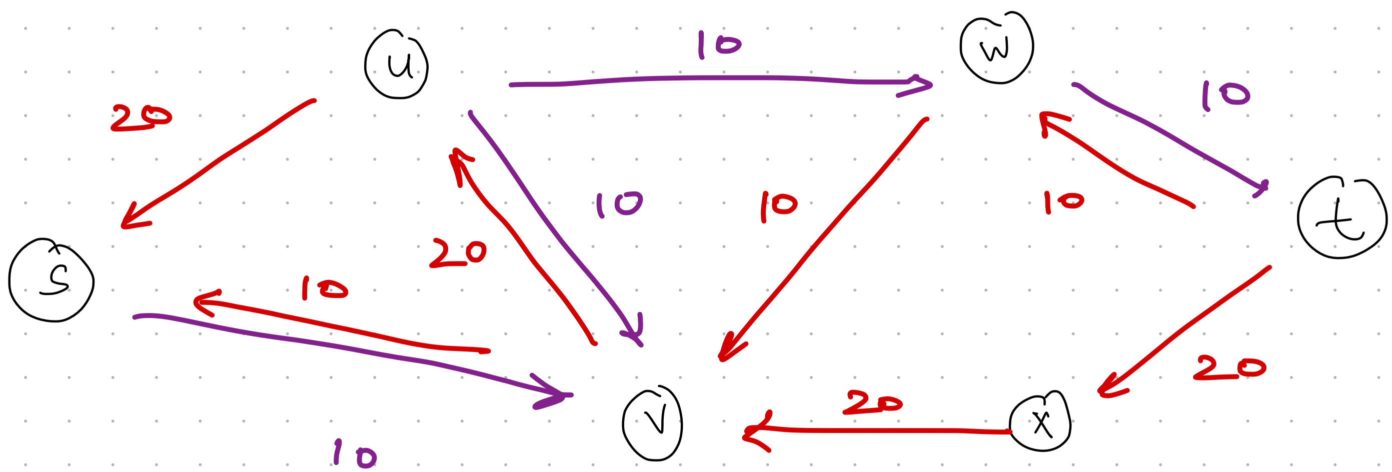


FORWARD RESIDUALS $f_e < c_e$

Output of Greedy

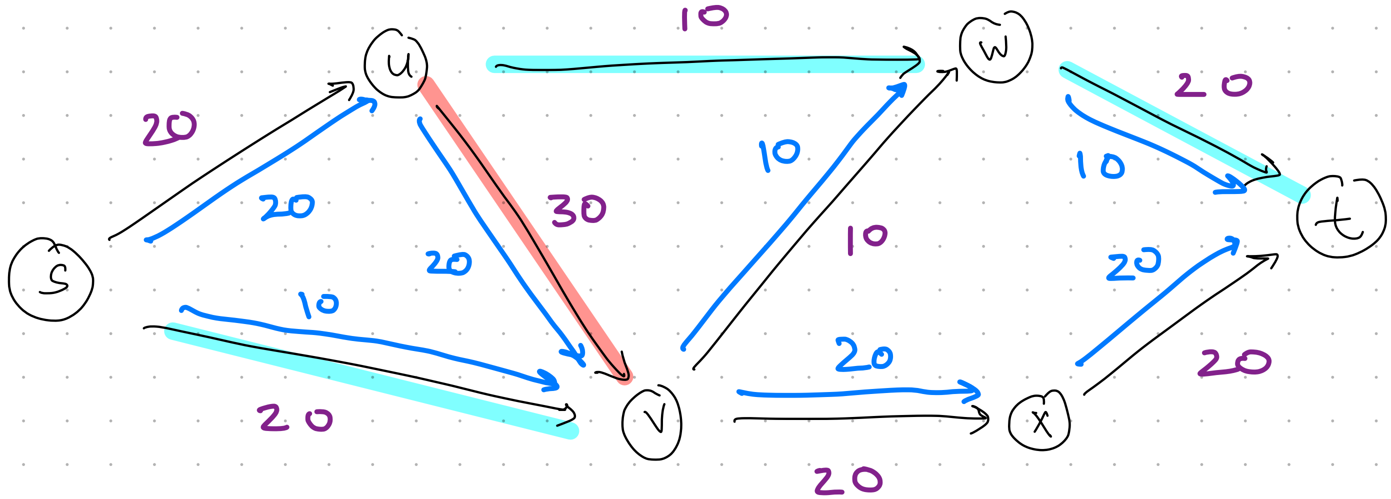


Residual Network

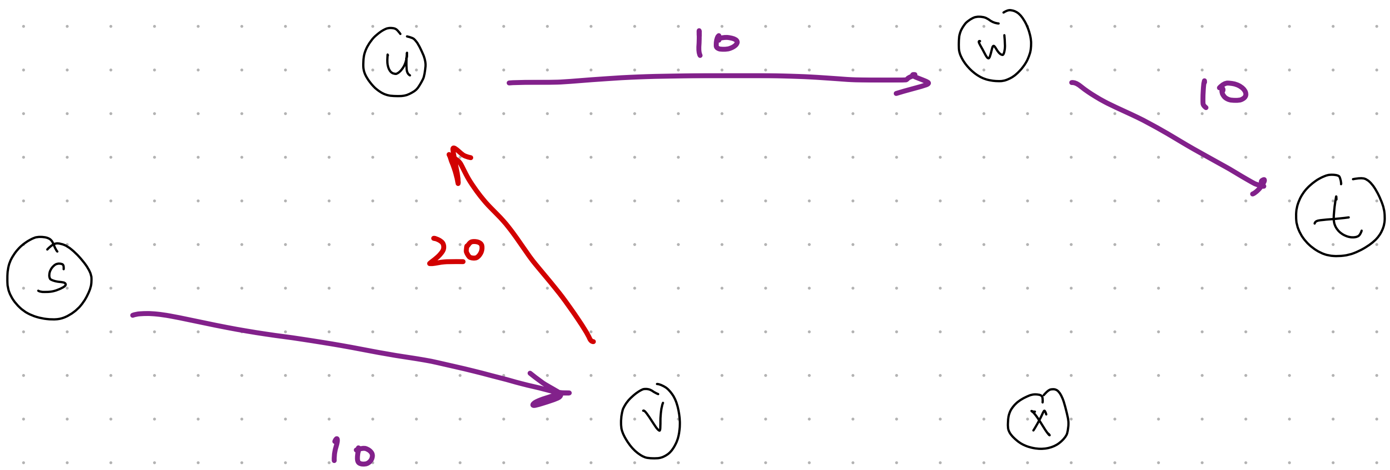


BACKWARD RESIDUALS $f_{\overleftarrow{e}} > 0$

Output of Greedy



Residual Network



st-path in Residual Network G^f ?

New Attempt. (Greedy in the Residual Network)

Initialize $f_e = 0 \quad \forall e \in G$.

Repeat.

New Attempt. (Greedy in the Residual Network)

Initialize $f_e = 0 \quad \forall e \in G$.

Repeat.

* Compute G^f the residual of current flow.

* if there exists an st-path in G^f "Augmenting path"
  - Push flow along path
  - Update flow f.

* if no st-path in G^f
  - Return f

New Attempt. (Greedy in the Residual Network)

Initialize $f_e = 0 \quad \forall e \in G$.

Repeat.

* Compute G^f the residual of current flow.

* if there exists an st-path in G^f "Augmenting path"
  - Push flow along path
  - Update flow f.

* if no st-path in G^f
  - Return f

Ford-Fulkerson Algorithm
for Max Flow.

Theorem. Ford-Fulkerson computes Max Flow correctly.

Theorem. Ford - Fulkerson computes Max Flow correctly.

* FF Returns a valid flow.
* Flow f^* returned by FF s.t.

$$\text{FlowOut}^{f^*}(s) \geq \text{FlowOut}^{f}(s)$$

for all valid flows f.

SIMPLIFYING ASSUMPTIONS

* capacities are integer

$$c_e : E \longrightarrow \mathbb{N}$$

* $|E| \geq \Omega(|V|)$

Theorem. Ford-Fulkerson computes Max Flow correctly.

* FF Returns a valid flow.

* Flow f^* returned by FF s.t.

$$\text{FlowOut}^{f^*}(s) \geq \text{FlowOut}^{f}(s)$$

for all valid flows f.

SIMPLIFYING ASSUMPTIONS

* capacities are integer

$c_e : E \longrightarrow \mathbb{N}$

* $|E| \geq \Omega(|V|)$

* if there exists an st-path in G^f

  - Push flow along path
  - Update flow f.

"Augmenting path"

Suppose st-path P has bottleneck capacity b in G^f

* if there exists an st-path in G^f

  - Push flow along path
  - Update flow f.

"Augmenting path"

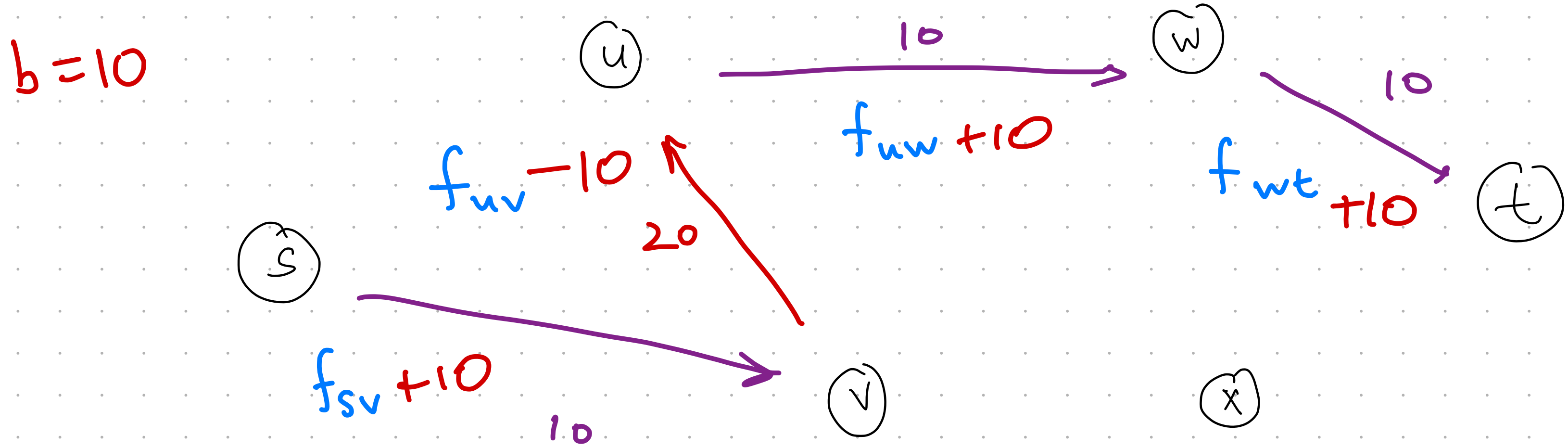
Suppose st-path P has bottleneck capacity b in G^f

For all forward residual edges in P.

$$f_e \leftarrow f_e + b$$

For all backward residual edges in P

$$f_e \leftarrow f_e - b.$$



For all forward residual edges in P.

$$f_e \leftarrow f_e + b$$

push more flow

For all backward residual edges in P

$$f_e \leftarrow f_e - b.$$

unpush flow

Lemma. Suppose P is an st-path through G^f with bottleneck capacity $b = \min_{e \in P} c^f(e)$.

Pushing b units of flow along P preserves CAPACITY & CONSERVATION constraints of f.

Lemma. Suppose P is an st-path through G^f with bottleneck capacity $b = \min_{e \in P} c^f(e)$.

Pushing b units of flow along P preserves CAPACITY & CONSERVATION constraints of f.

CAPACITY

Forward edges: $c_e - f_e = c^f(e)$

$$\Rightarrow c_e = f_e + c^f(e) \geq f_e + b \quad \blacksquare$$

Backward edges: $f_e - b \geq f_e - c^f(e) = f_e - f_e = 0 \quad \blacksquare$

Lemma. Suppose P is an st-path through G^f with bottleneck capacity $b = \min_{e \in P} c^f(e)$.

Pushing b units of flow along P preserves CAPACITY & CONSERVATION constraints of f.

CONSERVATION

For all $v \in V \setminus \{s,t\}$:

In-Edge. $\xrightarrow{\text{in}} (v) \xrightarrow{\text{out}}$ Out-Edge.

Forward

$\Delta \text{FlowIn}(v) = +b$

Backward

$\Delta \text{FlowOut}(v) = -b$

Forward

$\Delta \text{FlowOut}(v) = -b$

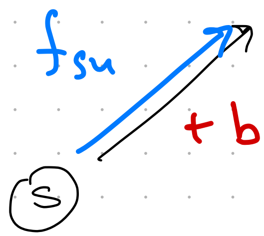
Backward

$\Delta \text{FlowIn}(v) = +b$

$\Rightarrow$ All in-out options preserve conservation. $\blacksquare$

So, FF preserves that f is a valid flow

Why does FF terminate?



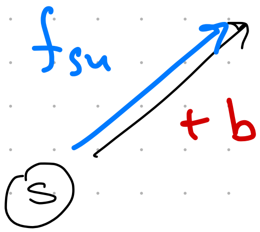
$\text{FlowOut}^{f}(s)$

$\downarrow$

$\text{FlowOut}^{f+b}(s)$

So, FF preserves that f is a valid flow

Why does FF terminate?



$\text{FlowOut}^{f}(s)$

$\downarrow$

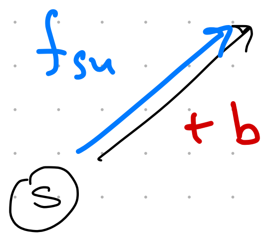
$\text{FlowOut}^{f+b}(s)$

Every augmenting path <u>increases</u> flow out of s

↳ By integer weights $b \geq 1$ in every iteration.

So, FF preserves that f is a valid flow

<u>Why does FF terminate?</u>



Every augmenting path <u>increases</u> flow out of s

$$\mathrm{FlowOut}^{f}(s) \downarrow \mathrm{FlowOut}^{f+b}(s)$$

↳ By integer weights $b \geq 1$ in every iteration.

$\Rightarrow$ After $\mathrm{Cap}(s) = \sum_{sv \in E} c_{sv}$ iterations, No more augmenting paths.

Ford-Fulkerson Running Time

Suppose $c^* = \max_{e \in E} c_e$.

$\Rightarrow \mathrm{Cap}(s) \leq c^* \cdot n$.

How long does each iteration take?

Ford-Fulkerson Running Time

Suppose $c^* = \max_{e \in E} c_e$.

$\Rightarrow \text{Cap}(s) \leq c^* \cdot n$.

How long does each iteration take?

* Construct G^f $\longleftarrow$ ≤ 2 edges per $e \in E$
* Find augmenting path P $\longleftarrow$ BFS in G^f from s
* Update flow f $\longleftarrow$ 1 update per edge in simple path

Ford-Fulkerson Running Time

Suppose $c^* = \max_{e \in E} c_e$.

$\Rightarrow \text{Cap}(s) \leq c^* \cdot n$.

How long does each iteration take?

* Construct G^f $\leftarrow$ ≤ 2 edges per $e \in E$
* Find augmenting path P $\leftarrow$ BFS in G^f from s
* Update flow f $\leftarrow$ 1 update per edge in simple path

$O(m) + O(m) + O(n)$

RT: $\boxed{O(c^* \cdot n \cdot m)}$ // pseudopolynomial time.