24 February 2025	Max Flow Algorithms:
· · · · · · · · · · · · · · · · · · ·	Ford - Fulkerson.
Plan	
* Ideas for Flow Algorit	turs
* Announcements	
* Ford - Fulkerson	
· · · · · · · · · · · · · · · · · · ·	
· · · · · · · · · · · · · · · · · · ·	
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·

Network Flow Problem.
\rightarrow Given Flow Networke G, s,t, c: $E \rightarrow \mathbb{R}^+$
\star Find Flow $f: E \rightarrow \mathbb{R}^+$ subject to
Capacity Constraints
$\forall e \in E$
$C_{e} = C_{e} = C_{e$
Conservation Constraints
Hve V ZS, tJ
$\sum_{uv \in E} f_{uv} = \sum_{vw \in E} f_{vw}$

Max Flow. Find Flow ft that maximizes Flow Out (s)

Attempt #0, Greedy * Find st - path w/ positive capacity * Push flow along path Repect until No sr Ŵ (,U) , 20 20 / 30 · · · · ∕ · **Ⅰ,○** · \hat{s} 20 \rightarrow (\vec{N}) ⇒(х) 0

Attempt #0, Greedy * Find st - path w/ positive capacity * Push flow along path Repect / until No sant Ś (u) ~ 20 20 20/2 $\left(\hat{S}\right)$ 20 \rightarrow $(\hat{\lambda})$ $\overrightarrow{}$ (\overrightarrow{x}) 20

Attempt #0. Greedy * Find st - path w/ positive * Push flow along path capacity Repect until No s~st Ś (u) 20/ $\left(\overline{\mathcal{N}} \right)$ ⇒ (x) 20 20 Ê) $(\underline{\mathbb{S}}) \xrightarrow{\longrightarrow} (\underline{\mathbb{V}}) \xrightarrow{\longrightarrow}$ \longrightarrow

Attempt #0. Greedy * Find st - path w/ positi * Push flow along path	re capacity Repect until No s~st
	\rightarrow (w) 20 20 4 10 10 10 10 10 10 10 10
$2 \circ \qquad (S) \rightarrow (O) \rightarrow (O)$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
No. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	. .

Attempt #0. Greedy * Find st - path w/ positive capacity * Push flow along path Repect " until No sant (W) (U) · ~ 20 S ·\0 20 X $\textcircled{\begin{tabular}{c} \hline \begin{tabular}{c} \hline \begi$ $() \longrightarrow () \longrightarrow (+)$

Attempt #0. Greedy * Find st - path w/ positive capacity * Push flow along path Repect " until No s~st paths (U) · ~ ·\0 $2 \sim (N)$ ſχ) $\textcircled{\begin{tabular}{c} \hline \begin{tabular}{c} \hline \begi$ $() \longrightarrow () \longrightarrow ()$

Attempt #0. Greedy * Find st - path w/ positive capacity * Push flow along path) Repect / unitil No s~st (\mathbf{u}) \mathbf{u} \mathbf{v} \mathbf{v} \mathbf{v} 10 30 30 · · · · · / 0 X) $(\bigcirc) \xrightarrow{} (\bigcirc$

Attempt #0, Greedy * Find st - path w/ positive capacity * Push flow along path Repect / unitil No s~st paths - (u) ~ (+) $(\bigcirc) \xrightarrow{} (\bigcirc$ $(1) \quad (2) \quad (2)$ No more s -> t paths GREEDY FLOW: 30

Problem ~ / Greedy Ivrevocable Myopica 10 Ψ Ű 20 10 20 30. Ó ŝ 20 10 20 Ö 3 (\mathbf{x}) Ń 7 O $\log(G) = 40.$ MaxFl

Problem w/ Greedy Ivrevocable Myopica 0 30 Ŝ 0 20 $\widehat{(\mathbf{v})}$ Max Flow(G) = 40. Greedy solution assigned for = 20 but . . f* = 10;

Announcements
* Prelim #1 Grades Released After Lecture
* HW #3 Released Wednesday,
* Recitation Stouts again on Schurday.
· ·

Problem w/ Greedy Ivrevocable Myopica Ψ 20 30. Ó 20 . \0 20 <u>⊰</u> (x) Ö 7 O Goal. Systematic way to "Revoke" on flow so far.

Augmenting a flow	· · · · · · · · · · · · · · · · · · ·
	· · · · · · · · · · · · · · · · · · ·
$\sum_{i=1}^{n} \frac{1}{2} $	
2.0	· · · · · · · · · · · · · · · · · · ·
2.0	
	· · · · · · · · · · · · · · · · · · ·
To conserve Flow at N.	
To conserve Flow at N.	$ \cdot \cdot$
To conserve $Flow at v$. $\Delta Flow In(v) = \Delta Flow Ont(v)$	$\left(\begin{array}{cccccccccccccccccccccccccccccccccccc$
To conserve $Flow at v$. $\Delta Flow In(v) = \Delta Flow Ont(v)$	$\left(\begin{array}{cccccccccccccccccccccccccccccccccccc$
To conserve $Flow$ at v . $\Delta Flow In(v) = \Delta Flow Ont(v)$	$\left(\begin{array}{cccccccccccccccccccccccccccccccccccc$
To conserve $Flow at v$, $\Delta Flow In(v) = \Delta Flow Ont(v)$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$

Augmenting a flow 20 To conserve Flow at $\Delta Flow In(v) = \Delta Flow Ont(v)$

G, source s, since t Given a Flow Network w/ capacities c: E > Rt and $f \models E \rightarrow \mathbb{R}^{t}$ G is a flow network The Residual Network v/ same vertex set as G. and residual capacities cf: E - Rt nueve Allow us to reason about flow we can still push/unpush C_{i}^{\prime} (uv) = $\left(\begin{array}{c} \mathbf{v} \\ \mathbf{v} \end{array} \right)$

Criver a Flow Network	G, source s, sink t w/ capacities $c: E \rightarrow R^{\dagger}$ and flow $f: E \rightarrow R^{\dagger}$
The Residual Network	G is a flow network v/ same vertex set as G.
and residual capacitie	s cf: E , Rt.
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$if f_{mv} = C_{mv}$
	at capacity

Criver a Flow Networ	The G_1 source s, since the weight of C_1 source s, since the weight C_2 and C
The Residual Netwo	vle G is a flow network vl same vertex set as G.
and residual capa	cities cf: E - Rt, where
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$if f_{wv} = C_{wv}$
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	FORWARD RESIDUAL
	v still has capacity to push flow

Criven a Flow Network	G, source s, since t w/ capacities c.E > R ⁺
The Residual Network	and flow $f \in \mathbb{R}^+$ G is a flow network
	v/ same ventex set as G.
and residual capaciti	es cf: E - Rt vhere
	$if f_{wv} = C_{wv}$
$C_{i} = C_{i} + C_{i$	$\frac{1}{2} + \frac{1}{2} + \frac{1}$
$C_{inv} = -f_{inv}$	BACKWARD RESIDUAL
WW K	as capacity to be "unpushed"

Output of Greedy	$\begin{array}{c} 10 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\$
Residual Network	(3)
	NO RESIDUAL $f_e = c_e$

Output of Greedy	$\begin{array}{c} 10 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\$
Residual	$(1) \qquad (10 - 0) \qquad (10) \qquad (10)$
	FORWARD RESIDUALS $f_e < c_e$

Output of Greedy	$\begin{array}{c} 10 \\ 10 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\$
Residual Network	$(u) \qquad 10 \qquad (w) \qquad 10 \qquad (v) \qquad $
. .	BACKWARD RESIDUALS ft > 0

Output of Greedy (S)	$ \begin{array}{c} 10 \\ 10 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 10 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 2$
Residual Network	$(u) \qquad 10 \qquad (w) \qquad 10 \qquad (e) \qquad $
st -path	No X N Residual Network GF?

New Attempt	C. (Greedy	in the	Residual	Network)
Initialize	$f_e = 0$	$fe \in C_{\ell}$	· · · · · · · ·
Repeat.	· · · · · · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · ·
· · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · · · · ·	· · · · · · · ·
· · · · · · · · · · · · · · · ·			· · · · · · · · · ·	· · · · · · · ·
· · · · · · · · · · · · · · · ·		· · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · ·
· · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · ·
· · · · · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · ·
· · · · · · · · · · · · · · ·		· · · · · · · · ·		· · · · · · · ·
· · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · · · · ·	· · · · · · · ·

New Attempt.	(Greedy	in the	Residual	Network)
Initialize f	$e^{-2} = 0$	fe e G	· · · · · · · · · · ·	· · · · · · · ·
Repeat	· · · · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · ·
* Compute y	the rest	idual of	current d	
* if there ex	cists an	st-path	r in the Gt and	
Push flow	s along	path		
- Update	flow f		Augun	ective a
* if no st -	flow f. path in Gf		Augun	eiting him in a second
* if no st - Return	flow fr		Aigui	
* if no st - (- Return	flow f .		Aiguit	
× if no st (- Return	flow f .			

New Attempt. (Greedy in the Residual Network)
Initialize fe= 0 HeeG.
Repeat.
* Compute 4 the residual of current flow
* if there exists an st-path in Gt
- Push flow along path - Update flow f. Auguerting
* if no st-path in Gf
Return f
Ford - Fulkenson Algorithm for Max Flow,
· · · · · · · · · · · · · · · · · · ·

Theorem	Ford - Fulkerson	computes	Max Flow	correctly.
		• • • • • • •		
		••••••		

Theorem	Ford - Fulkerson	computes Max Flow	correctly.
· · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·
$\begin{array}{c} & & \\$	Returns a valid f* returned by	$f \leftarrow v$	<td< th=""></td<>
· · · · · · · · · ·	Flow Ontf	*(s) <u>></u> Flow Outf(s)	l valid flows f.
· · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·		· · · · · · · · · · · · ·
		SIMPLIFYING ASS	SUMPTIONS
· · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	* Capacities Ce:E-	ave integer
· · · · · · · · · ·		\times	$\sum_{i=1}^{n} \left(\sum_{i=1}^{n} \left(\sum_{i$

Theorem	Ford - Fulkerson	computes Max Flow	correctly.
· · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · ·
× FF Re	turns a valid		· · · · · · · · · · · ·
F = F + F = F + F + F + F + F + F + F +	* veturned b	y FFF s.t	
· · · · · · · · · · ·	Flow Out	$f^{*}(s) \rightarrow f(s)$	· · · · · · · · · · ·
· · · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	for al	l valid flows f.
· · · · · · · · · · · ·	· · · · · · · · · · · · · ·		· · · · · · · · · · ·
· · · · · · · · · · ·		SIMPLIFYING ASS	JUMPTIONS
· · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	* capacities	ave integer
	· · · · · · · · · · · · · ·		$\sum_{i=1}^{n} \left(\left(\sqrt{i} \right)^{i} \right)^{i} = \left(\sqrt{i} \right)^{i} =$

× if	there exists an st-path in Gf
· · · · · · · · · · · · ·	Push flow along path
	Update flow t Auguerting path.
· · · · · · · ·	$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot &$
· · · · · · · · ·	bottlenech capacity b in Gf
· · · · · · · ·	
· · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
	· · · · · · · · · · · · · · · · · · ·
	· · · · · · · · · · · · · · · · · · ·
· · · · · · · ·	
· · · · · · · ·	. .

* if there exists an st-path in Gf - Push flow along path - Update flow f Augmenting path Suppose st-path P has bottlenech capacity b in Gf For all forward residual edges in P $f_e \leftarrow f_e +$ For all backward residual edges in P $f_e \leftarrow f_e \leftarrow f_e$

$\mathbf{b} = 10$	$(u) = \frac{10}{10}$ $f_{uv} + 10$ 20	W f w t to t
· · · · · · · · · · · · · · · · · · ·	$f_{sv} + 10$	
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	all forward vesidual $f_e \leftarrow f_e + b$	edges in P push more flow
	all backward residual fe < fe b	eders in P unpush flow
· · · · · · · · · · · ·		. .

	Suppose F with bot	is an st-pa Henech capacit	th through G^{f} the = min $C^{f}(e)$
Pushing	be units	of flow along	Preserves
CAPP	$f_{1}(1)$	CONSERVATION	constraints of f.
· · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · ·
· · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · ·	· · · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · ·	· · · · · · · · · · ·		
· · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · ·	· · · · · · · · · ·		· · · · · · · · · · · · · · · · ·
· · · · · · · · · ·			· · · · · · · · · · · · · · · · · · ·

Lemma. Suppose P is an st-path through Gf with bottlenech apacity b = min cf(e) eEP	•
Pushing & units of flow along P preserves CAPACITY & CONSERVATION constraints of f	•
$\frac{CAPACITY}{Forward edges}: C_e - f_e = c^{f}(e) \\ \Rightarrow c_e = f_e + c^{f}(e) \ge f_e + b \mathbb{Z}$	•
Backward edges: $fe - b \ge fe - c^{f}(e) = fe - fe = 0$	· · ·
	· · · ·

Lemma. Suppose P is an with bottlenech a	st-path through Gf apacity b = min cf(e) eEP
Pushing & units of flow a CAPACITY & CONSERVA	along P preserves ATION constraints of f.
CONSERVATION For all NEV 125, t3.	. .
$\frac{\text{In}-\text{Edge}}{\text{Forward}} \xrightarrow{\text{in}} (v) \xrightarrow{\text{out}} \\ \Delta Flow In(v) = + b$	$Q_{1}t - Edge$. Forward $\Delta Flow O_{1}t(v) = -b$
Backword $\Delta Flow Out(v) = -b$	Backward $\Delta Flow In(v) = + b$
>> All in-out options preser	ve concervation.

So, FF preserves	that f is a valid flow
Why does FF	terminate?
. .	
f s s s s s s s s s s s s s s s s s s s	· · · · · · · · · · · · · · · · · · ·
	· · · · · · · · · · · · · · · · · · ·
FlowOrtf (s)	
	. .
Flow Out (s)	
· · · · · · · · · · · · · · · · · · ·	
· · · · · · · · · · · · · · · · · ·	
	· · · · · · · · · · · · · · · · · · ·

So, FF preserves that f is a valid flow	· · · · ·
Why does FF terminate?	
· · · · · · · · · · · · · · · · · · ·	· · · · ·
for the Every augmenting path	
S increases flow out of	· · · · · ·
Flow O + f(s)	· · · · ·
Li By integer weights b ≥	
Flow Out (s) in every iteration	••••••
· · · · · · · · · · · · · · · · · · ·	· · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · ·

So, FF preserves that f is a valid flow	•
Why does FF terminate?	•
· · · · · · · · · · · · · · · · · · ·	•
For th Every augmenting path increases flow out of s	•
FlowOwtf(s)	•
Li By integer weights b ≥ 1 Flow Out ^{f+b} (s)	•
\Rightarrow After $Cap(s) = \sum_{sv} c_{sv}$ iterations, No more svete paths.	· · ·

Ford-Fulkerson Running	Time .
Suppose c* = max Ce eEE	. .
\Rightarrow $Cap(s) \leq c^* N$	
Hand Jona dees each	iteration tako?
· · · · · · · · · · · · · · · · · · ·	
· · · · · · · · · · · · · · · · · · ·	<pre></pre>
· · · · · · · · · · · · · · · · · · ·	
· · · · · · · · · · · · · · · · · · ·	. .
 	
	· · · · · · · · · · · · · · · · · · ·

Ford -	Fulkerson	Runnize	Time	· · · · · ·	· · · · · · · · · · · · ·
Suppose	$c^* = wc$	xx C _e E		· · · · · ·	· · · · · · · · · · · · ·
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\alpha p(s) \leq 1$		· · · · · · · · ·	· · · · · ·	· · · · · · · · · · · · ·
	Long does		evetion t	ako 7	· · · · · · · · · · · · · · ·
× ((,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	Construct	Cf.	· · · · · · · · · · · ·		<2 edges per
		,			eet
	Finda	gmenting	path P	· · · · · · · ·	BFS in Gf
	Find av Update	flow f	Path P		BFS in Gf from S I update
	Find av Mpdate	flow f	Paff + P + P + P + P + P + P + P + P + P +		BFS in Gf from S I update per edge in simple path
	Findav	flow f	Path P		BFS in Gf from S I update per edge in simple path
	Findav	flow f	· path · p · · · · · · · · · · · · · · · · ·		BFS in GF From S I update per edge in Simple path

Ford-Fulkenson Running Time
Suppose c* = max Ce
$\Rightarrow (c_{\alpha}p(s)) \leq c_{\alpha} N_{\alpha},$
How long does each iteration take?
* Construct Gf eEE
* Find augmenting path P <- BFS in Gf From S
× Update flow f
simple path.
O(w) + O(w) + O(w)