

Lecture 16

Cuts and Flows in Graphs

Q. Given a directed graph G , and vertices s, t , is there a directed path from $s \rightarrow t$ in G ?

Ans :

For the next few lectures :

Q. Given a directed graph G , vertices s, t , are

there at least k -paths from $s \rightarrow t$ that do not share edges?

One natural scenario:

G represents a telecommunication network, so, we would like robustness in case of failure.

We will see :- efficient algorithms for this problem,

And some very remarkable applications.

Crucial objects: cuts and flows in a graph.

Flows

Directed graph G:

Flow on G: weight assignment to the edges satisfying certain constraints

Pictorially: G represents a network of pipes and there is a liquid flowing through it (water, oil, data) etc.

① There is an external source of the liquid \rightarrow a source vertex, from where all the liquid is coming through. (denote by s)

And

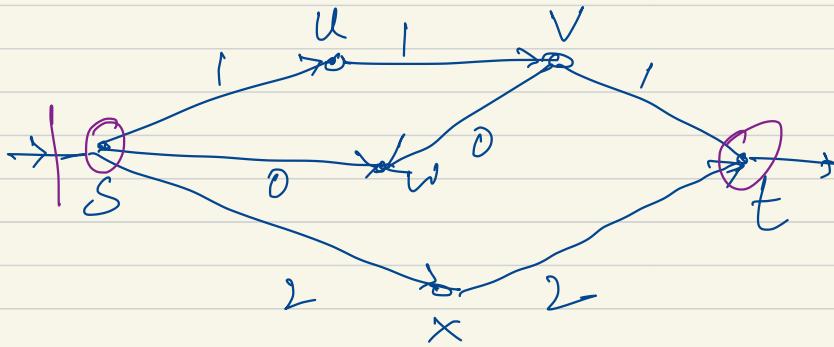
② A sink through which everything is being flushed out. (denoted by t)

③ The supply is continual.

Natural properties

① Individual fibres might have some properties.

② there is no accumulation of liquid at any vertex.



Formally

Flow Network

$(G, s, t, u) : G \rightarrow$ directed graph
 $s, t \rightarrow$ designated source
and sink resp

$u : E \rightarrow \mathbb{R}_{\geq 0} \rightarrow$ capacity
of each edge

Feasible flow/valid flow

Valid flow on a flow network is a
function : $f : E \rightarrow \mathbb{R}_{\geq 0}$ satisfying the following.

- ① (Capacity) For every $e \in E$, $f(e) \leq u(e)$ }
 $f(e) \geq 0$ ✓

② (conservation) For every $v \in \{s, t\}$

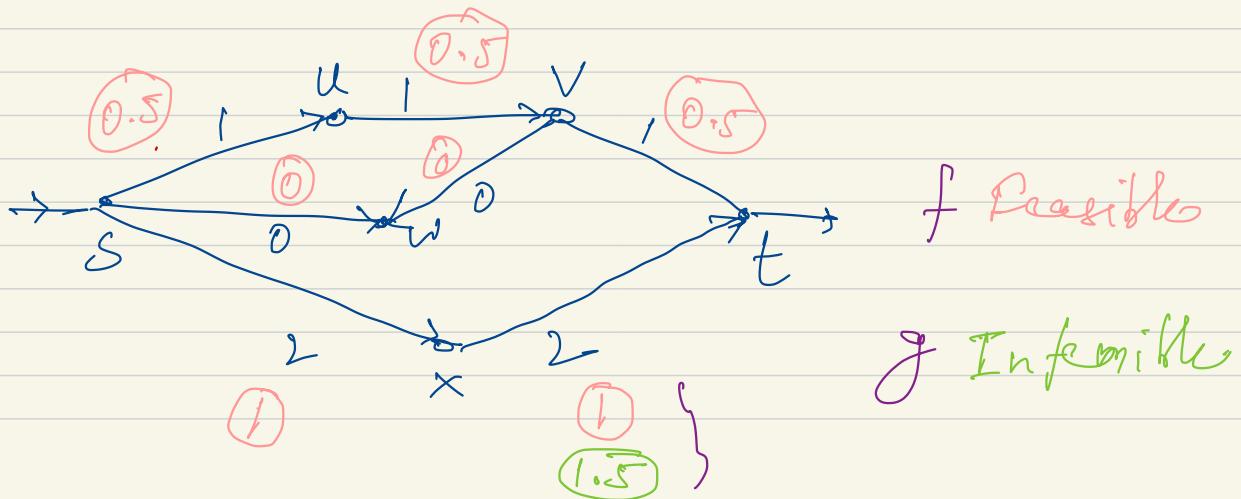
$$\sum_{(u, v) \in E} f(u, v) = \sum_{(v, w) \in E} f(v, w)$$



incoming



outgoing



Excess of a given flow (G, s, t, u) - flow_{net}

$f: E \rightarrow \mathbb{R}_{>0}$ - (arbitrary fn.
not necessarily a valid
flow)

$\text{Excess}_f: V \rightarrow \mathbb{R}_{>0}$ a function

$$\text{Excess}_f(v) = \sum_{(x, v) \in E} f(x, v)$$

$$(x, v) \in E$$

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

$$(v, w) \in E$$

(Incoming - Outgoing)

Rate of accumulation at a vertex

Lemma 1:

Given any $f: E \rightarrow \mathbb{R}_{>0}$

$$\sum_{v \in V} \text{Excess}_f(v) = 0 .$$

$$\text{Pf}^0 = \sum_{v \in V} \left(\sum_{(x, v) \in E} f(x, v) - \sum_{(v, w) \in E} f(v, w) \right)$$

$$= \sum_{(\alpha, \beta) \in E} \left(\underbrace{f(\alpha, \beta)}_{\text{at } \beta} - \underbrace{f(\alpha, \beta)}_{\text{at } \alpha} \right)$$

\leftarrow 0.

17

Feasible flow:

- ① capacity constraints hold
- ② excess at any non-source
non-sink vertex equals 0.

Lemma 1 \Rightarrow ③ $\text{Excess}_f(\text{sink})$

$$= -\text{Excess}_f(\text{source}).$$

Value of flow

Val of a valid flow f

$\hat{=} \text{ Total excess at the sink}$

$\hat{=} - \text{Total excess at the source}$

Note: - flow is a directed object initially.

- can view on undirected graph as having edges in both directions.
- we will work with directed graphs here.

Max-Flow Problem:

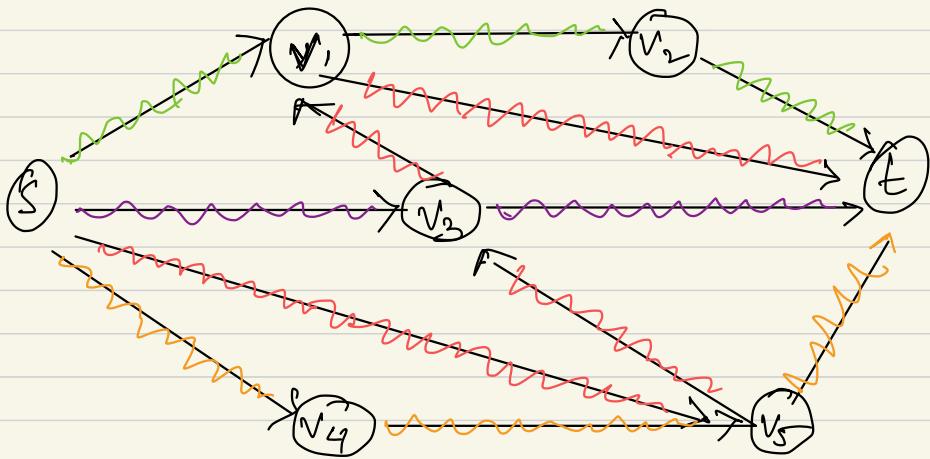
Input: A flow network (G, s, t, u)

Output: A feasible flow of max value

(how much liquid can we
push through the network)

Flow vs Disjoint path problem that we
started with.

Capacity = 1
on each edge



4 - edge disjoint s-t paths.

Claim. Can send a flow of value at least 4
from s to t.

Why? Assign a flow of value 1 to each edge.

In general:

Lemma 2 If all edge capacities are 1 and there are k disjoint paths from s - t , then the max flow value is at least k .

Proof: Let P_1, P_2, \dots, P_k - are the k-edge disjoint paths from $s \rightarrow t$ in G .

$$f: E \rightarrow \mathbb{R}_{\geq 0} \quad \left\{ \begin{array}{ll} \text{if } e \text{ is on any of the P's,} \\ f(e) = 0 \quad \text{o/w.} \end{array} \right.$$

Check feasibility

↗

other direction

Lemma 3 If there is a valid flow of rank k with $\text{fle}(\cdot) = \underline{0}/1$, then, there are k edge disjoint $s \rightarrow t$ paths.

Proof: $\tilde{G} - Q$ with all edges of weight 0 deleted.

Claim: t is reachable from s in \tilde{G} .

- Let P_1 be a $s \rightarrow t$ path in \tilde{G} .
- Define $\tilde{G}_2 \leftarrow \tilde{G} \setminus P_1$

claim: \tilde{G}_2 has a flow of rank $k-1$. In fact the same flow we started with values — ◻
Induction on $k-1$ —

Slightly uncomfortable issue: — assuming $f(e) = 0 / 1$ for every e .
— Later: this is without loss of generality.

Thus: on unit capacity graphs, an algorithm for max flow problem solves the disjoint path problem that we started with.

Cuts in a graph

Disjoint path problem:

max number of edge disjoint paths between $s \rightarrow t$ in G .

(Measure of robustness of the network G)

Min cut problem: min number of edges that can be deleted such that in the remaining graph, there are no $s \rightarrow t$ paths?

(seems closely related to the former question
, at least intuitively.)

- More generally, in a weighted graph, we want to delete edges such that their sum of weights is smallest.

Formally,

A cut:

- Flow network (G, s, t, u)

- $F \subseteq E$ is an s - t cut if there is no

path from s-t in $G \setminus F$.

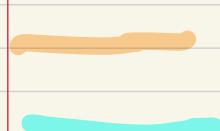
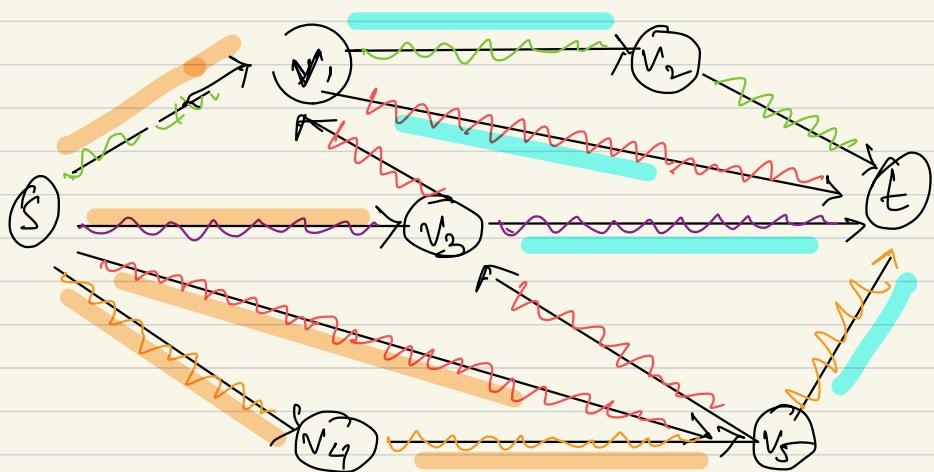
Capacity of a cut

$$\text{cap}(F) := \sum_{e \in F} u(e)$$

Minimal s-t cut

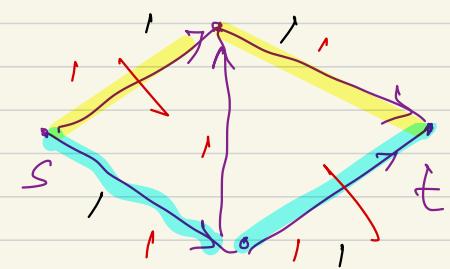
Cut F is minimal if any strict subset

$F' \subsetneq F$ is not an s-t cut



Y cut?

Minimel cuts?



One more useful definition

Out boundary

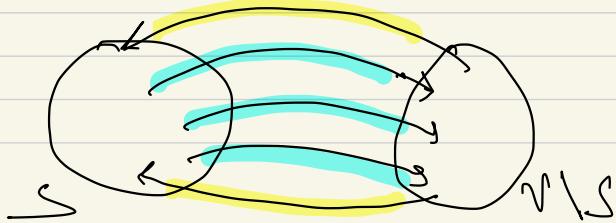
$G = (V, E)$ - directed graph

$S \rightarrow$ source vertex
 $S \subseteq V$
subset

$$S \subseteq V$$

Out boundary (S): $\partial^+(S)$

$$\partial^+(S) := \left\{ (x, y) \in E : x \in S, y \notin S \right\}$$



Similarly, in boundary.

$$\partial^+ S = \left\{ (x, y) \in E : x \notin S, y \in S \right\}$$

Minimal cuts vs out boundaries

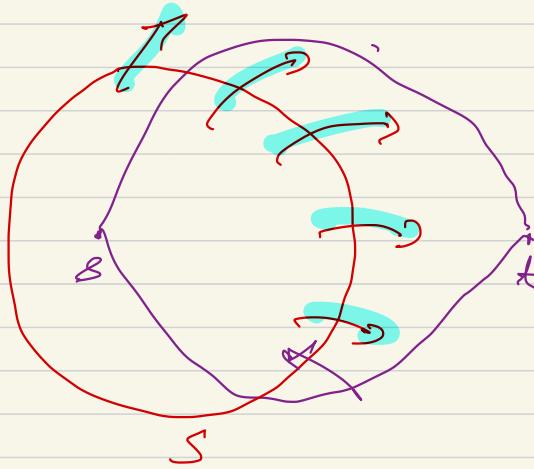
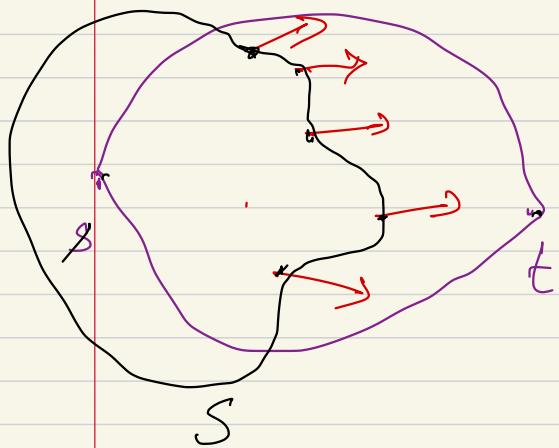
Lemma:

① For every $S \subseteq V$ s.t $s \in S, t \notin S$
 $\partial^+ S$ is an $s-t$ cut.

② Any minimal $s-t$ cut

if the out boundary of some $S \subseteq V$
with $s \in S, t \notin S.$

Example:

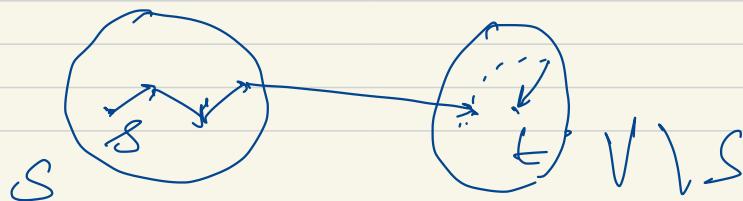


(Note: not precise converses of each other)

Why?

Proof: Fix any $S \subseteq V$

① $s \in S, t \notin S$



Take any s - t path.

It intersects δ^+ 's somewhere ---

(2)

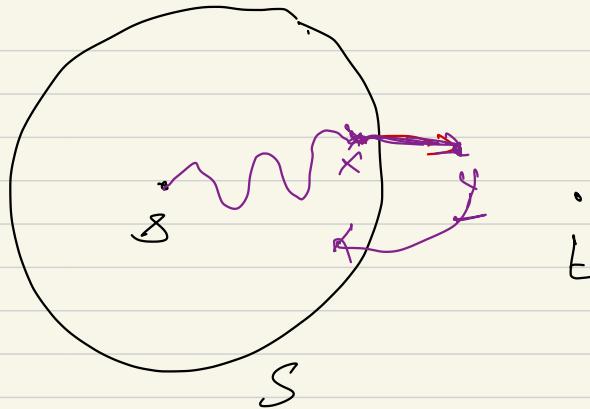
$P \subseteq E$ — minimal s - t cut.

$S = \{w \in V : w \text{ is reachable from } s$
 $\text{in } G \setminus P\}$

$t \notin S$

(a) $\partial^+ S \subseteq F$

Why?



(b) $P \subseteq \partial^+ S$

From (a) $\partial^+ S \rightarrow S-t\text{-cut}$

From (a) $\partial^+ S \subseteq P$

Minimality of P.

Min s-t cut problem

In put: Flow network (G, s, t, u)

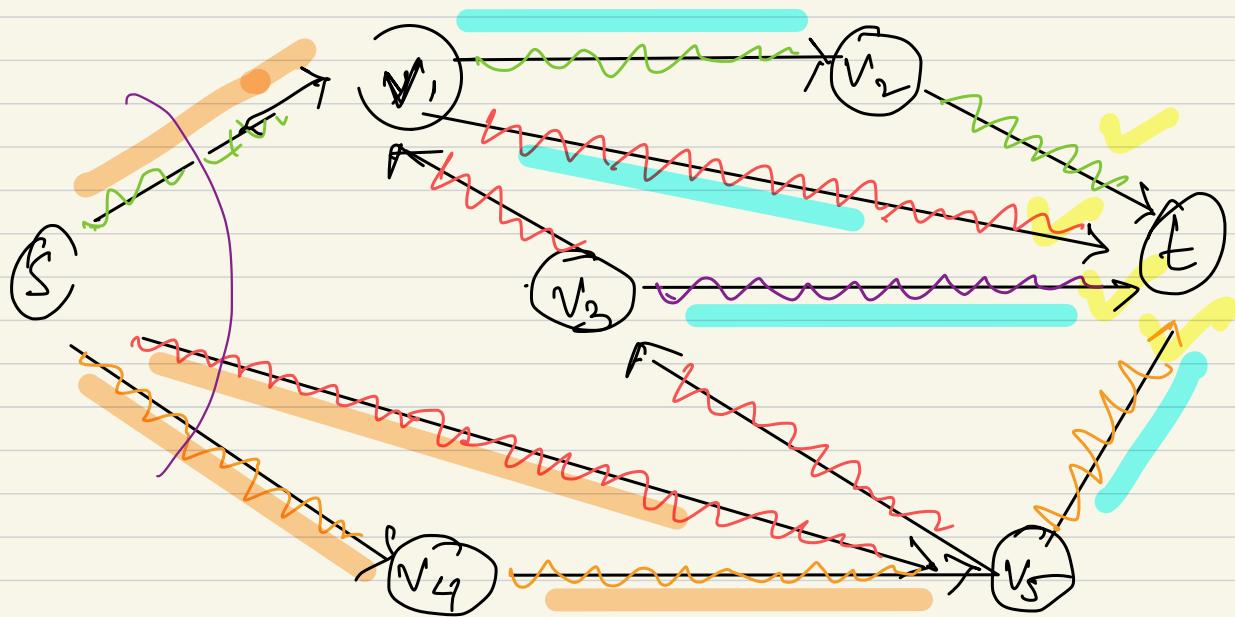
Output: s-t cut of minimum capacity.

In other words,

$$\min_{S \subseteq V} \text{cap}(\delta^+ S).$$

$$S \leftarrow \emptyset$$
$$t \notin S$$

$$\text{cap}(\delta^+ S) = \sum_{e \in \delta^+ S} u(e)$$



Duality of flows and cuts.

Max s-t flow } two sides of
Min s-t cut } the same coin

Plan

- ① Formalize what this means
- ② See the proofs of these connections

C Weak duality (gamma for max flow-min cut)

Lemma (G, s, t, u)

Let f - any feasible $s-t$ flow

δ^+S - any $\{s-t\}$ cut
minimal.

Then,

$$\boxed{\text{val}(f)} := \text{Excess}_f(t) \leq \boxed{u(\delta^+S)}$$

$$\text{where;} u(\delta^+S) = \sum_{e \in \delta^+S} u(e)$$

(Weak duality theorem)

[Integro edge
capacities]

Theorem: In any flow network

$$\text{max s-t flow} \leq \text{min s-t cut capacity}$$

Will also show a stronger theorem.

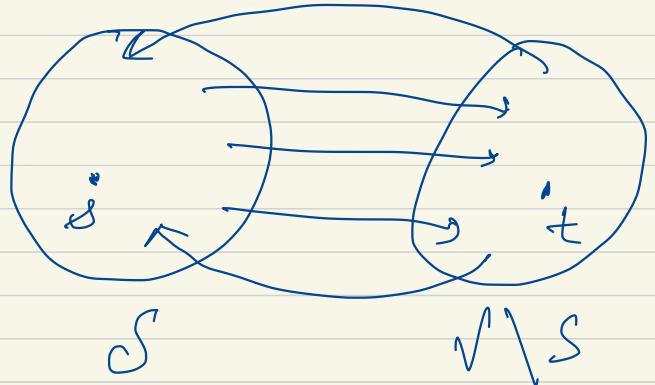
Theorem (Max flow - min cut theorem)

$$\text{max s-t flow} = \text{min s-t cut capacity}$$

Proof of Lemma:

$\text{val}(f)$

= amount of flow
coming into s



By Feasibility:

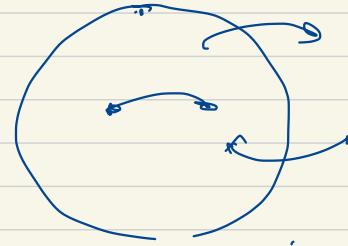
— All this flow has to go out of set S to reach the sink t in $V \setminus S$.

— so, the flow has to use edges in $\delta^+ S$ to go out.
— so, $\text{val}(f) \leq u(\delta^+ S)$

Formally:

Let us bound

$$\sum_{v \in S} \text{Excess}_f(v)$$



① From point of view of vertices.

$$\text{Excess}_f(u) = 0 \quad \forall v \notin s, t \quad \left. \begin{array}{l} \text{ } \\ \text{ } \end{array} \right\} \text{of feasibility.}$$

$$\Rightarrow \sum_{v \in S} \text{Excess}_f(v) = \text{Excess}(S)$$

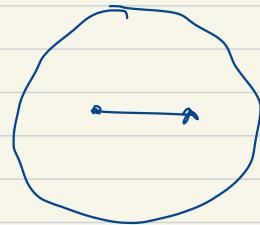
② From point of view of edges.

$$\sum_{v \in S} \text{Excess}[v] = \sum_{v \in S} \left(\sum_{(x,v) \in E} f(x,v) - \sum_{(v,w) \in E} f(w,v) \right)$$

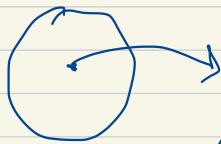
Three kinds of edges involved

① Both end points in S

Total contribution?



②

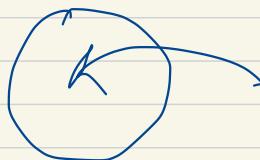


an edge in $\partial^+ S$

contribution $\Leftarrow -f(e)$

(3)

an edge in $\delta^+ S$



contribution = $f(e)$

$$\text{So, } \sum_{e \in \delta^+ S} f(e) = \sum_{e \in \delta^+ S} f(e)$$

$$= \sum_{e \in \delta^+ S} f(e)$$

Note

$$\begin{aligned} \text{val}(f) &= -\text{Excess}(\varsigma) \\ &= \sum_{e \in \delta^+ \varsigma} f(e) - \sum_{e \in \delta^- \varsigma} f(e) \end{aligned}$$

Recall: $\begin{cases} \textcircled{1} \quad f(e) > 0 \\ \textcircled{2} \quad f(e) \leq u(e) \end{cases} \quad \{ \neq e$

$$\text{So, } \text{val}(f) \leq \sum_{e \in \delta^+ \varsigma} f(e)$$

$$\leq \sum_{e \in \delta^+ \varsigma} u(e) \leq u(\delta^+ \varsigma)$$

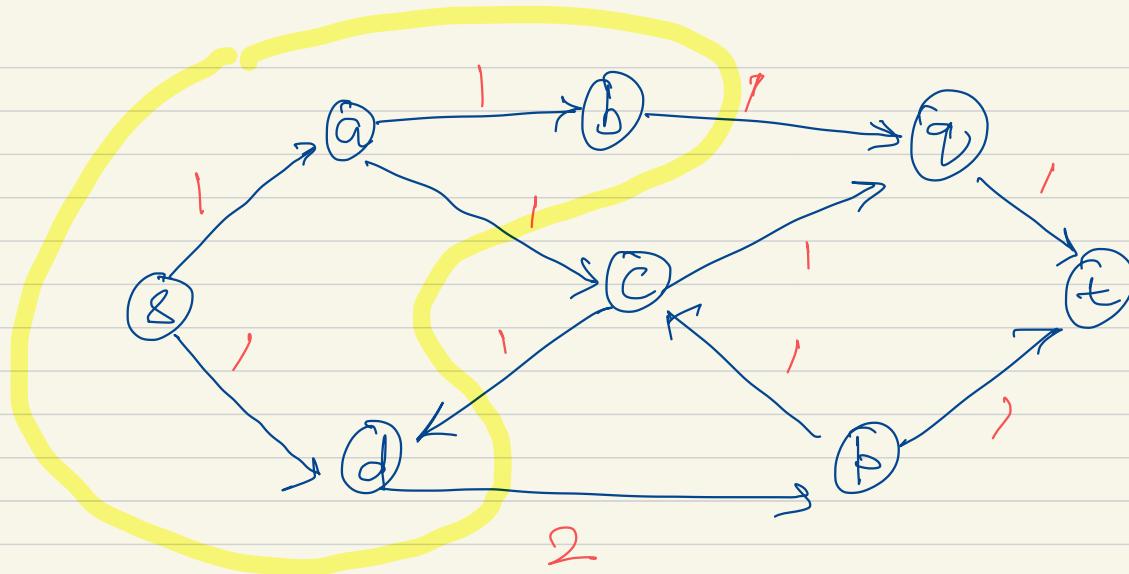
②

Immediati consequence:

(Weak duality theorem)

Theorem: In any flow network

$$\text{max s-t flow} \leq \text{min s-t cut capacity}$$



All capitals are 1

When is the inequality in the Lemma an equality.

Theorem:

Suppose $f \rightarrow$ feasible s-t flow

S is a s-t cut such that

$$\textcircled{1} \quad f(e) = u(e) \quad \forall e \in \delta^+ S$$

$$\textcircled{2} \quad f(e) = 0 \quad \forall e \in \delta^- S$$

Then,

- f is a maximum s-t flow, and

- S is a minimum s-t cut, and

their values are the same.

Pf:

When are the inequalities in the proof of Lemma equalities?



Next: Strong duality or Max-flow min-cut theorem.

Theorem:

$$\begin{aligned} & \text{max s-t flow value} \\ &= \text{min s-t cut capacity} \end{aligned}$$

- Proof will be via an algorithm of Ford and Fulkerson from the 50's.
- Will need to discuss and build up some prelims before describing the algorithm.

Residual Networks

Goal: find a valid flow that maximizes
 $\text{Excess}_f(t)$.

Idea:

- ① Start with zero flow on every edge
(will increase the flow in iterations)

② Find an s-t path ρ in G (use BFS/
DFS - —)

③ let $\delta = \min_{e \in \rho} u(e)$

④ For every $e \in \rho$, set $f(e) = f(e) + \delta$

(flow augmentation)

(increased the flow by δ , remains
a valid flow - why?)

Want to repeat this procedure -- how do
we proceed in the next iteration?

Algo 1

1) start with $f(e) = 0 \forall e$ → current flow

$$u_f(e) = u(e) \quad \text{→ current capacity}$$

② while true do:

find any path p from $s \rightarrow t$

$$\text{s.t } \min_{e \in p} u_f(e) = \delta > 0$$

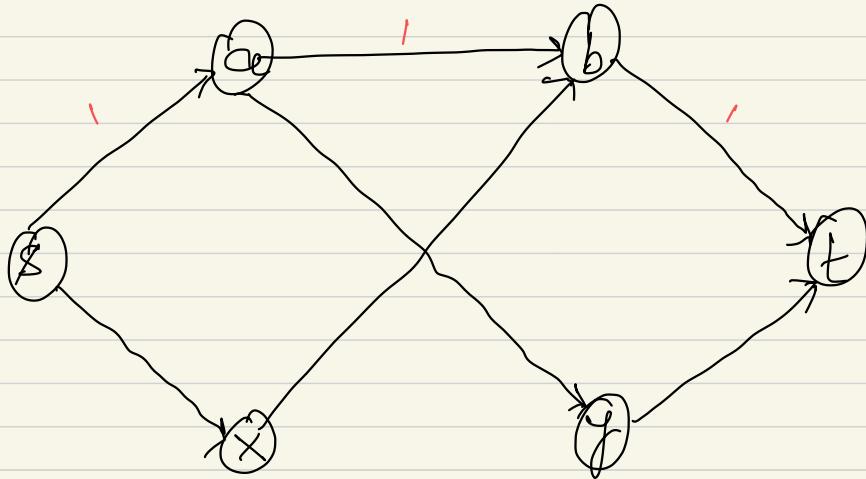
If no such path, stop and return f

for every $e \in p$, $f(e) \leftarrow f(e) + \delta$

$$u_f(e) \leftarrow u_f(e) - \delta$$

③ Return f .

- At every stage $u_f(e) + f(e) = u(e)$.
- Ensure that the flow is feasible at every stage of this algo.
- Does this output the max flow?



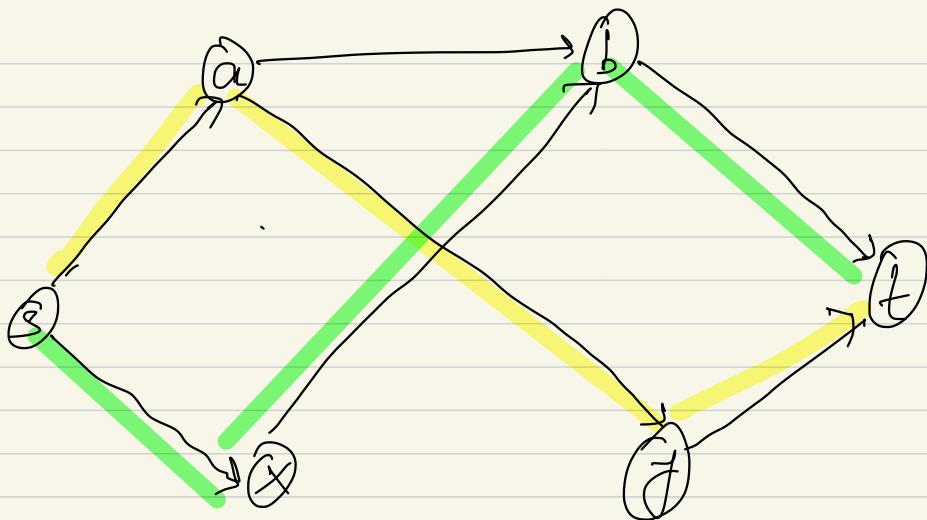
$$cc(c) = 1$$

$\neq e$

$$p_1 = s \rightarrow a \rightarrow b \rightarrow t \quad \delta = 1$$

No further augmentation --

What is the max flow?



What went wrong?

- Bad choice of the initial path?

Residual Networks: lets us correct the incorrect choices we have made in the past iterations.

Residual Network.

Given: (1) flow network (G, s, t, u)

(2) Valid flow $f: E \rightarrow \mathbb{R}_{\geq 0}$

Residual network with respect to flow f ,
denoted by G_f is defined as follows:

1) $G_f = (V, E_f)$ where $E_f = E \cup \bar{E}_{rev}$

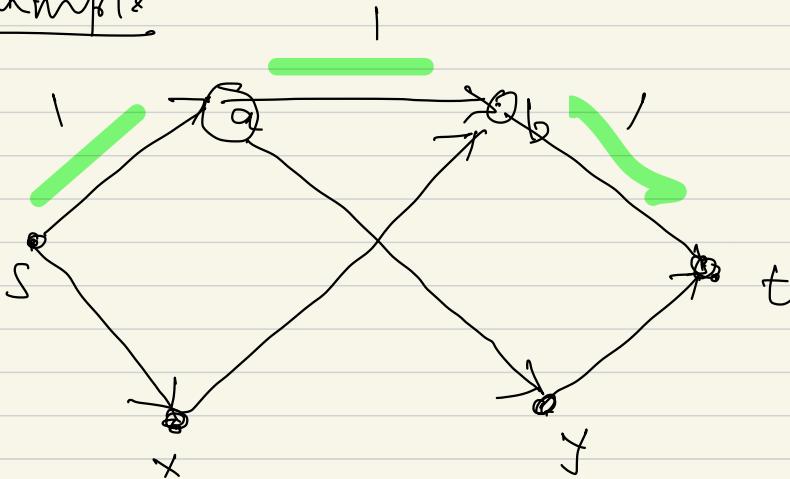
2) Residual capacities on the edges

$$u_f(x, y) = \begin{cases} u(x, y) - f(x, y) & \text{if } (x, y) \in E \\ f(y, x) & \text{if } (x, y) \in \bar{E}_{rev} \end{cases}$$

E_{rev} = set of reversal of edges in E

$$= \{ (y, x) \mid (x, y) \in G \}$$

Example:



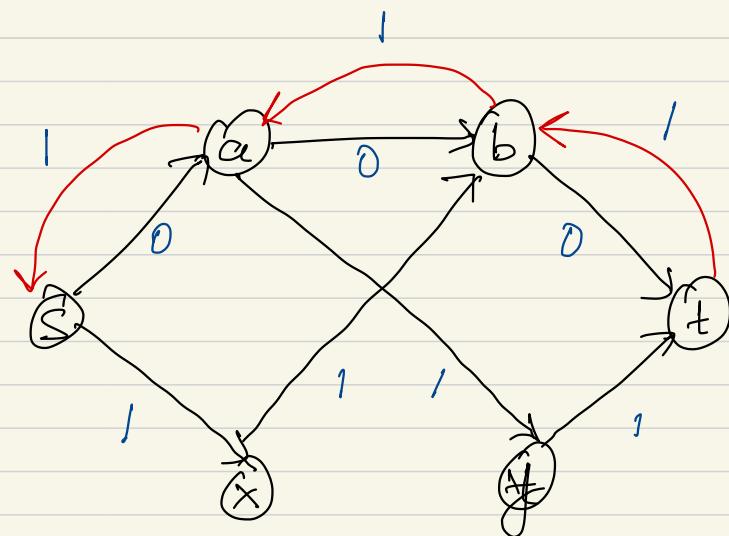
unit capacity

$f(e) = 1$

G, f

Residual Network

(ignoring cap 0
red edges)

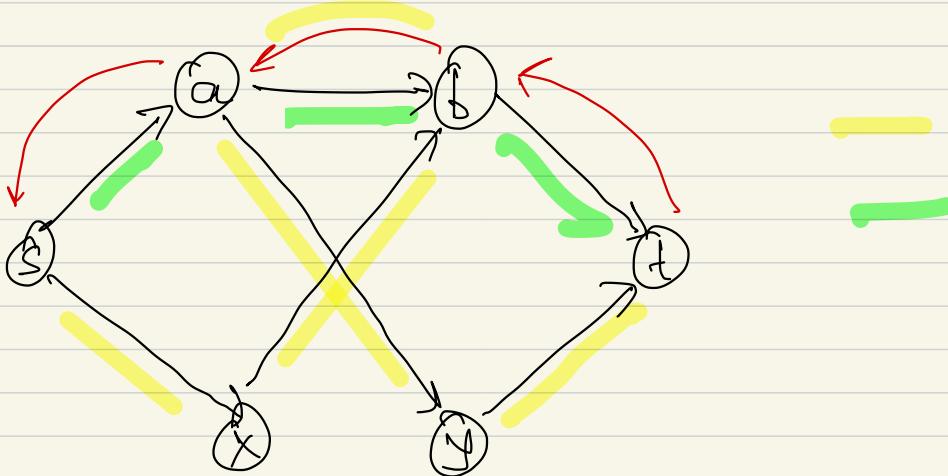


- Edges in the reverse direction give us a way of correcting the past mistakes.
- Intuitively: increasing flow along a ~~red edge~~ is like decreasing the flow in the opposite direction that is already there.

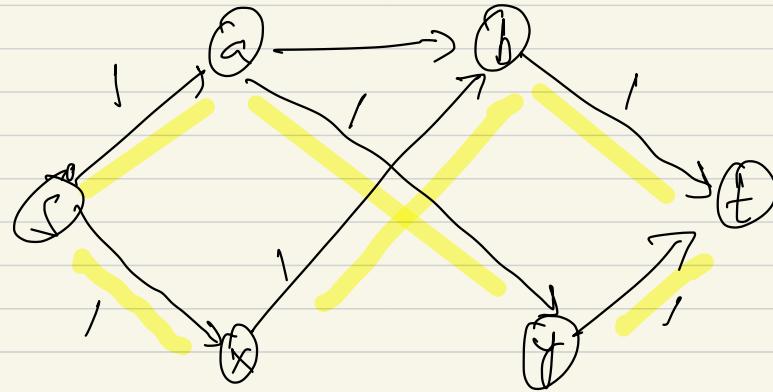
Ex: In G_f $s \rightarrow x \rightarrow b \xrightarrow{a} q \rightarrow y \rightarrow t$
is a path with y for each edge by 1.
We can augment the flow by 1 along this path.

Recall: $a \rightarrow b$ had flow 1 unit
Now, we want to push 1 unit of flow from
 $b \rightarrow a$. \Rightarrow Net flow from $a \rightarrow b = 0$.
(undoing the previous
assignment)

leads to the following



In the original graph



Indeed the max flow

For the next iteration, edge capacities etc have
to be updated appropriately.

Augmentation procedure (G_f, c, s, t, p)
 $\xrightarrow{p \rightarrow \text{short path in } G_f}$

① $f = \min_{e \in P} u_f(e)$

② For each $e \in P$,

if $e \in E$, $f(e) \leftarrow f(e) + \delta$

if $(x, y) \in E_{rev}$ (i.e. $(y, x) \notin E$)

$f(e) \leftarrow f(e) - \delta$.

③ Return f .

Properties of augmentation procedure

- Let f' be the new flow obtained by augmenting f along a path ρ .

Lemma 1: f' is a valid flow.

Pf: Capacity constraint + conservation constraint

- Only changes in edges on the path ρ .

① $e \in E$ → holds since $u_f(e) + f(e) = u(e)$
(original edge) by construction of u_f
and $f \leq u_f(e)$.

② $e = (x, y) \in E_{\text{par}}$ by design $u_f(e) = f(x, y)$
(i.e. $(y, x) \in E$) and $f \leq u_f(e)$.

So, $0 \leq f'(e) \leq u(e)$ \nparallel^2 .

Conservation constraints?

— — — —



Lemma 2

$$\text{val}(f') = \text{val}(f) + \delta$$

Pf:

$$\text{val}(f') = \text{Excess}_{f'}(t)$$

Look at the edge of b incident to t .

a) $e \in E$ \rightarrow

(b) $e \in E_{\text{excl}}$

for d-Fulkerson Algorithm

1) Initialise $f(es) = 0 \forall e$

2) while true, do :

(a) construct G_f, u_f

(b) If there is an s-t path p in G_f

s.t $\min_{e \in p} u_e > 0,$

- Augment (G_f, G, s, t, p) ,
to get the augmented flow f' .

- $f \leftarrow f'$

(c) Else Return f, r'

Correctness of Ford-Fulkerson

Lemma 1

If $u(e)$ is an integer for every $e \in E$
then,

1) FF terminates in $O(n \cdot m \cdot V)$ time

where $V = \max_{e \in E} u(e)$

2) FF outputs an integer valued
Valid flow f (f(e) \in \mathbb{Z}_{\geq 0})

Proof:

- We start with an integer valued flow.
 $f(e) \geq 0 \forall e$
- In every augmentation call, increase is integral.
- Total val of max flow $\leq nU$. (Why?)
- Hence at most nU augment calls.
- Each round takes about $O(n+m)$ time.



Lemma 2

FF returns the max flow correctly.

How do we prove maximality of the flow?

recall,

Theorem 1 f - feasible s-t flow.

S - s-t cut

such that,

$$a) f(e) = u(e) \quad \forall e \in \delta^+ S$$

$$b) f(e) = 0 \quad \forall e \in \delta^- S$$

Then, f is a maximum s - t flow, S is a min s - t , and their values are the same.

Proof of Lemma 2

Will describe a cut S that satisfies conditions of theorem above.

$S = \{v : v \text{ is reachable from } s$
 $\text{in } G_f \text{ with all zero capacity}$
 edges removed.

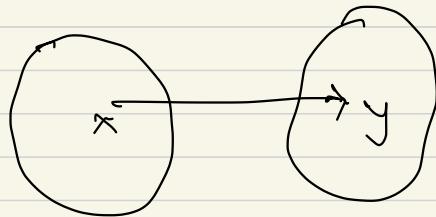
- 1) $x \in S$ { why?
 2) $t \notin S$ }

Want to argue that S satisfies conditions of the theorem ~~*~~

① Let $(x, y) \in \partial^+ S$

Then, $u_f(x, y) = 0$ { why?

$$\Rightarrow f(x, y) = u(x, y)$$



② $(x, y) \in \partial^- S$

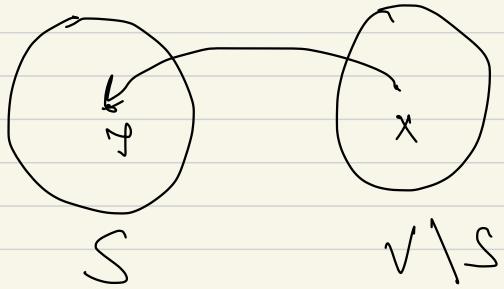
claim: $f(x, y) = 0$

If not, then, $f(x, y) > 0$

$\Rightarrow u_f(y, x) > 0$

\Rightarrow in $G_f \rightarrow$ (also contains reversed edges)
 x is reachable from y .

contradiction:



Next up:

Applications of Max-flow - min cut theorem
to a bunch of problems that do not
always look like questions about cuts and
flows.

