Cuts and Flows in Graphs

Q: Given a directed gph G, and vertices s, t, is there a directed path from $s \to t$ in G?

Ans:

For the next few lectures:

Q: Given a directed gph G, vertices s, t, are there at least k-paths from s→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

Picture: 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 — a source vertex, from where all the liquid is coming through. (denote by s)

And

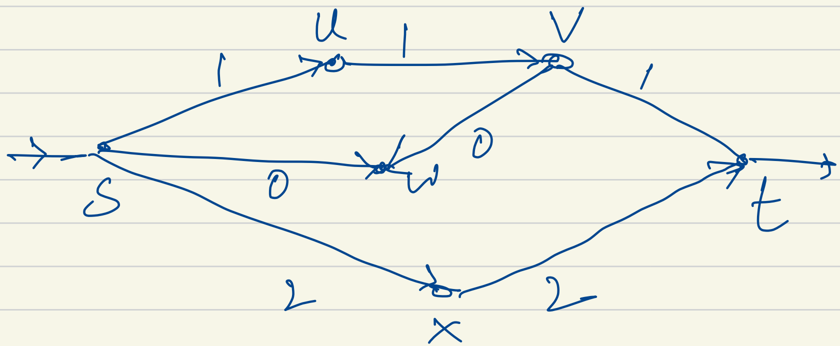
② A sink through which everything is being flushed out. (denoted by t)

③ The supply is continuous.

Natural properties

① Individual pipes might have some capacities.

(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

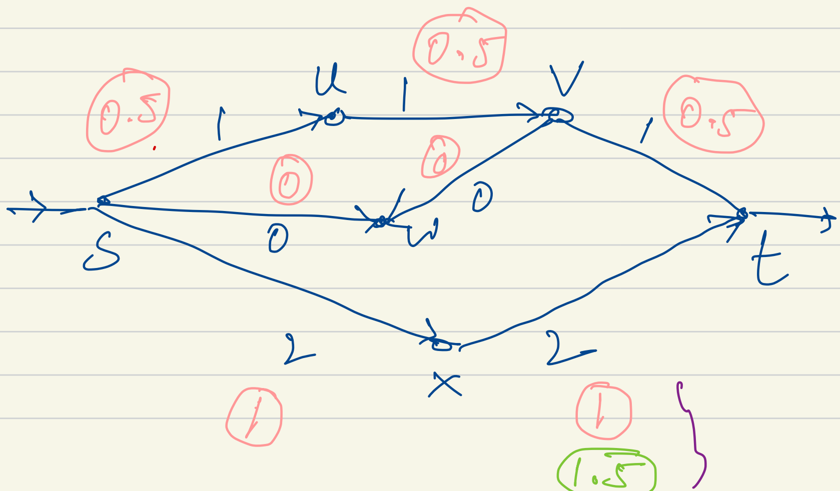
Feasible flow / valid flow

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 $v \notin \{s, t\}$

$$\underbrace{\sum_{(u,v)\in E} f(u,v)}_{\text{incoming}} = \underbrace{\sum_{(v,w)\in E} f(v,w)}_{\text{outgoing}}$$



f Feasible

g Infeasible

Excess of a given flow

(G, s, t, u) – flow net

$f : E \to \mathbb{R}_{\geq 0}$ – (arbitrary fn. not necessarily a valid flow)

$\text{Excess}_f : V \to \mathbb{R}_{\geq 0}$ a function

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

(Incoming – Outgoing)

Rate of accumulation at a vertex

Lemma 1:

Given any $f : E \to \mathbb{R}_{\geq 0}$

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

Pf:

$$\sum_{v \in V} \left(\sum_{(x,v) \in E} f(x,v) - \sum_{(v,w) \in E} f(v,w) \right)$$

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

$= 0.$

$\boxed{/}$

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

$:=$ Total excess at the sink

$:= -$ Total excess at the source

Note: - flow is a directed object intuitively.

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

Max-flow Problem:

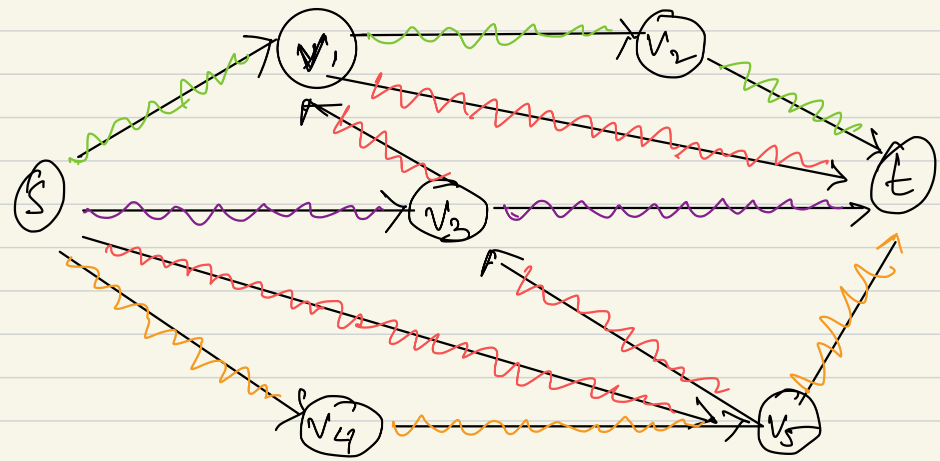
Input: A flow network (G, s, t, u)

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 ?

In general :

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

Proof :

4

Other direction

Lemma 3 If there is a valid flow of val k with $f(e) = 0/1$, then, there are k edge disjoint s-t paths.

Proof:

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 $s \to t$ 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 $s \to t$ 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

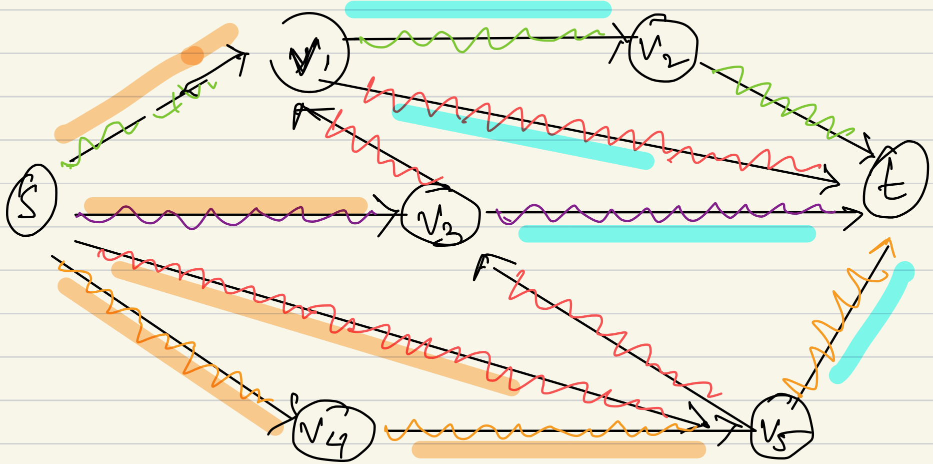
path from s-t in $G \setminus F$.

<u>Capacity</u> <u>of a cut</u>

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

<u>Minimal s-t cut</u>

cut F is minimal if any strict subset $F' \subsetneq F$ is not an s-t cut



} cuts ?

minimal cuts ?

One more useful definition

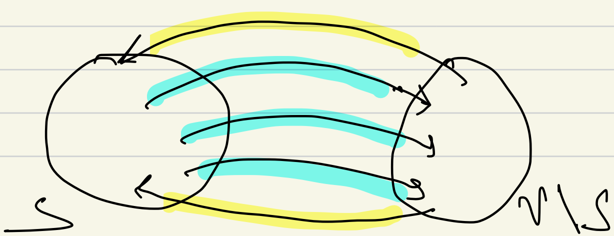
Out boundary

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

$S \subseteq V$

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

$:= \{ (x, y) \in E : x \in S, y \notin S \}$



Similarly, in boundary.

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

Minimal cuts vs out boundaries

Lemma:

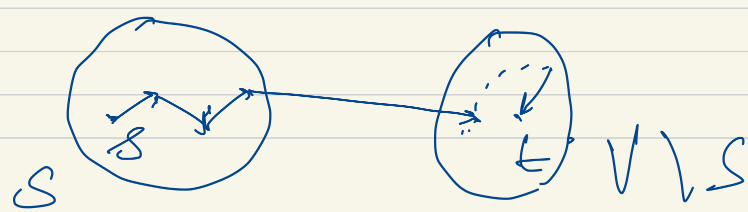
1. For every $S \subseteq V$ s.t $s \in S, t \notin S$ $\partial^+ S$ is an s-t cut.

2. ~~A~~ Any minimal s-t cut

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

(Note: not precise converses of each other)
Why?

<u>Proof</u>: Fix any $S \subseteq V$
①
$s \in S$, $t \notin S$



Take any s-t path.
It intersects $\partial^+ S$ somewhere - - -

(2)

$F \subseteq E$ — minimal s-t cut.

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

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

why?

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

from ① $\partial^+ S$ — s-t cut

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

Minimality of F.

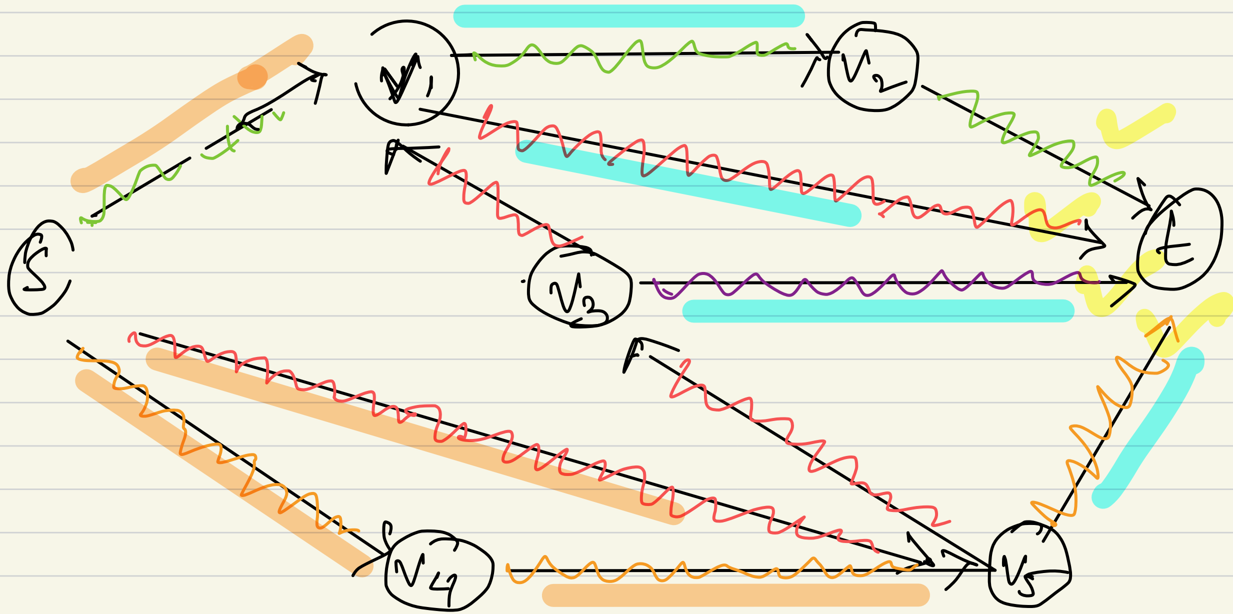
Min s-t cut problem

Input: Flow network (G, s, t, u)

Output: s-t cut of minimum capacity.

In other words,

$$\min_{S \subseteq V} \operatorname{cap}(\partial^+ S).$$



s
v_1
v_2
v_3
v_4
v_5
t

Duality of flows and cuts.

Max s-t flow
Min s-t cut

$\left.\begin{array}{l}\text{Max } s\text{-}t \text{ flow}\\ \text{Min } s\text{-}t \text{ cut}\end{array}\right\}$ two sides of the same coin

Plan

1. Formalize what this means
2. see the proofs of these connections

Lemma (G, s, t, u)

Let f – any feasible s-t flow

$\partial^+ S$ – any s-t cut

Then,

$$\mathrm{val}(f) := \mathrm{Excess}_f(t) \leq u(\partial^+ S)$$

where, $u(\partial^+ S) = \sum_{e \in \partial^+ S} u(e)$

(Weak duality theorem)

<u>Theorem:</u> In any flow network

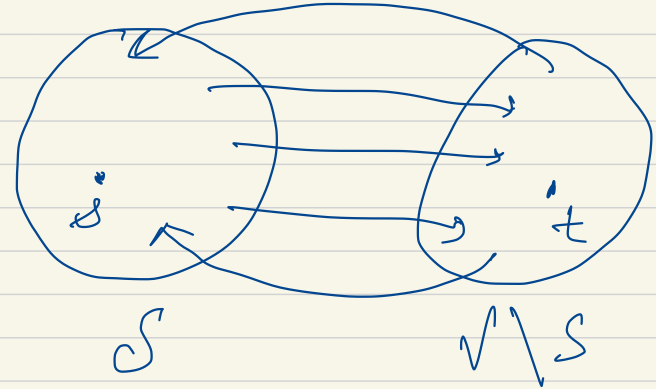
max s-t flow $\leq$ min s-t cut capacity.

Will also show a stronger theorem.

<u>Theorem</u> (Max flow - min cut theorem)

max s-t flow $=$ min s-t cut capacity

Proof of Lemma:



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 \ S.

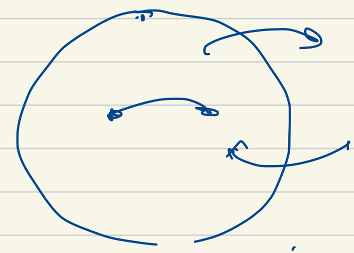
- so, the flow has to use edges in $\partial^+ S$ to go out.

- So, $val(f) \le u(\partial^+ S)$

Formally:

Let us bound

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



① From point of view of vertices:

$$\text{Excess}_f(v) = 0 \quad \forall v \neq s, t \quad \Big\}\ \text{feasibility.}$$

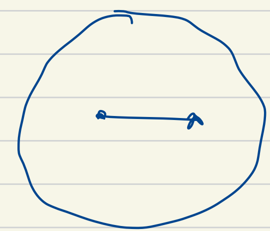
$$\Rightarrow \sum_{v \in S} \text{Excess}_f(v) = \text{Excess}(s)$$

(2) From point of view of edges.

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

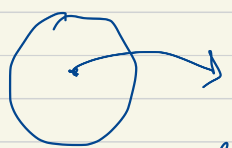
Three kinds of edges involved

(1) Both end points in S



Total contribution? 0

(2)

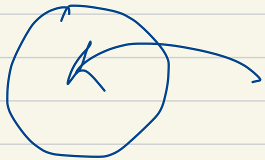


S

an edge in $\partial^+ S$

contribution $= -f(e)$

③ an edge in $\partial^- S$



contribution $= f(e)$

So, $\displaystyle\sum_{v \in S} \text{Excess}_f(v) = \sum_{e \in \partial^- S} f(e) - \sum_{e \in \partial^+ S} f(e)$

Now

$$\mathrm{val}(f) = -\mathrm{Excess}(S)$$

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

Recall. ① $f(e) \geq 0$ ② $f(e) \leq u(e)$ $\Big\}\ \forall e$

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

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

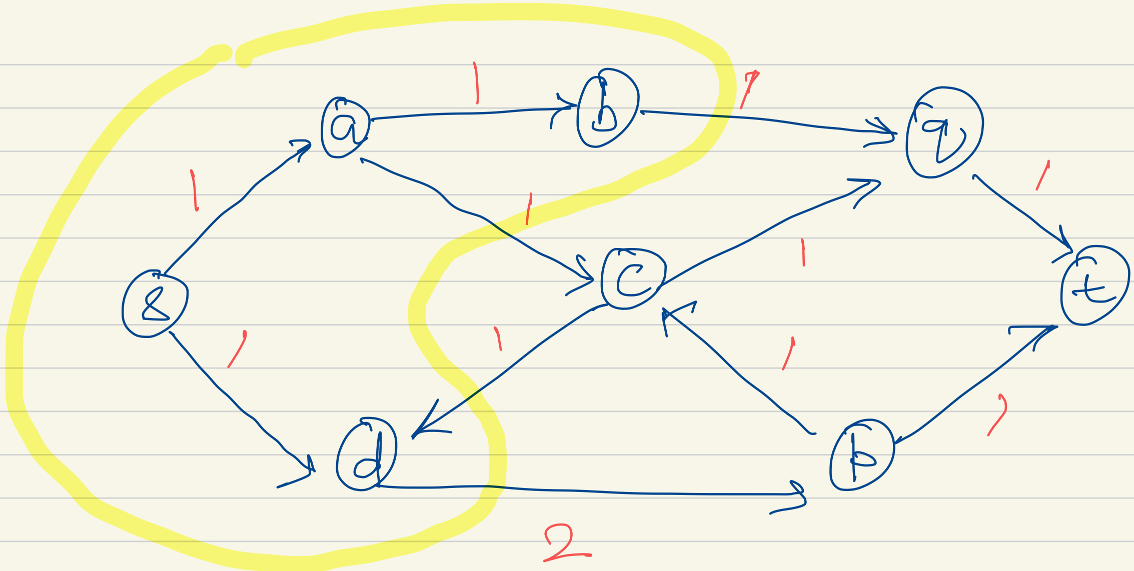
$\boxed{//}$

Immediate consequence:

(Weak duality theorem)

Theorem: In any flow network

max s-t flow $\leq$ min s-t cut capacity.



All capacities are 1

When is the inequality in the Lemma an equality.

Theorem:

Suppose $f \to$ feasible s-t flow

S - is an s-t cut such that

1. $f(e) = u(e) \quad \forall\, e \in \partial^{+} S$
2. $f(e) = 0 \quad \forall\, e \in \partial^{-} S$

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?

$\boxtimes$

Next: strong duality or max-flow min-cut theorem.

Theorem:

max s-t flow value
= min s-t cut capacity.

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

Residual Networks

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

Idea:

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

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

③ Let $\delta = \min_{e \in p} u(e)$

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

(flow augmentation)

(increase 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 $f(e) = 0 \quad \forall e$ → current flow

$u_f(e) = u(e)$ → current capacity

2) while true do:

find any path p from $s \to t$

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

If no such path, stop and return f

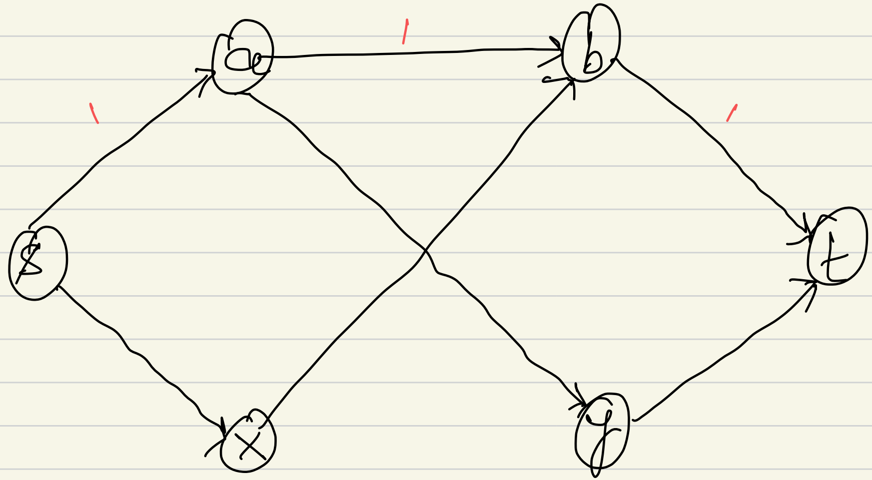
for every $e \in p$, $f(e) \leftarrow f(e) + \delta$

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

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?

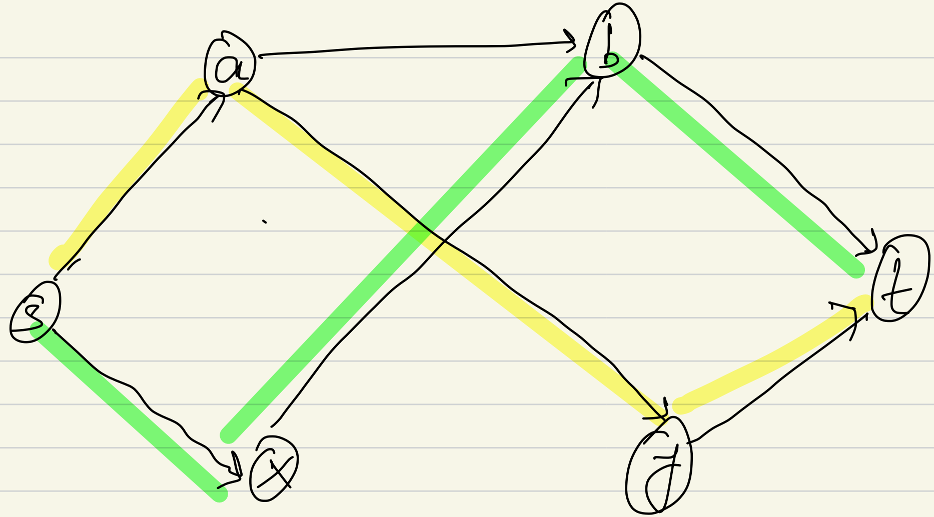
$c(e) = 1$
$\forall e$



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

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 $f: E \to \mathbb{R}_{\geq 0}$

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

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

2) Residual capacities on the edges

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

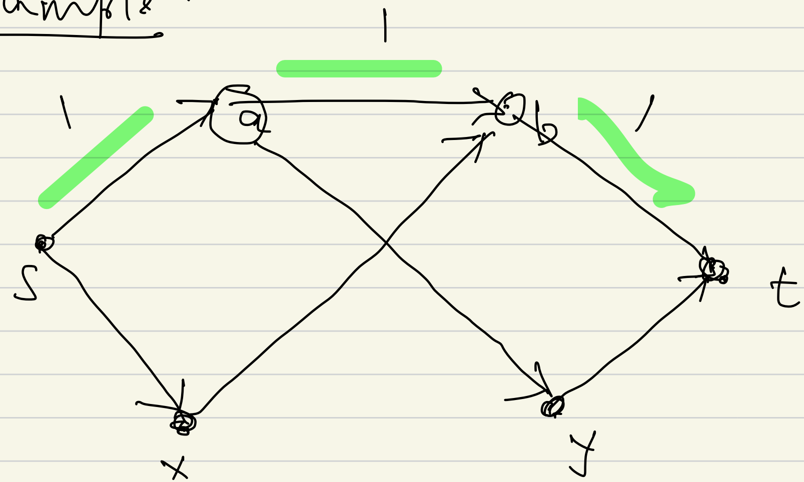
E_{rev} = set of reversal of edges in E

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

Example:

unit capacity

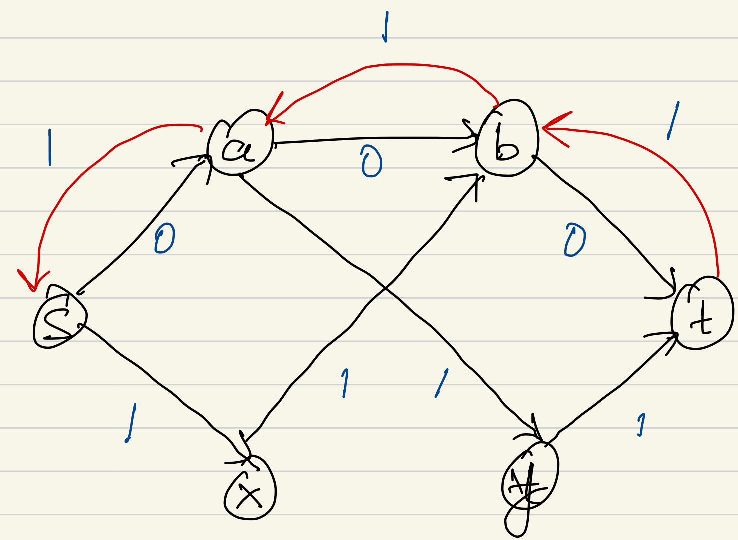
— $f(e) = 1$



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 $\quad s \longrightarrow x \longrightarrow b \longrightarrow a \longrightarrow y \longrightarrow t$

is a path with u_f for each edge being 1.

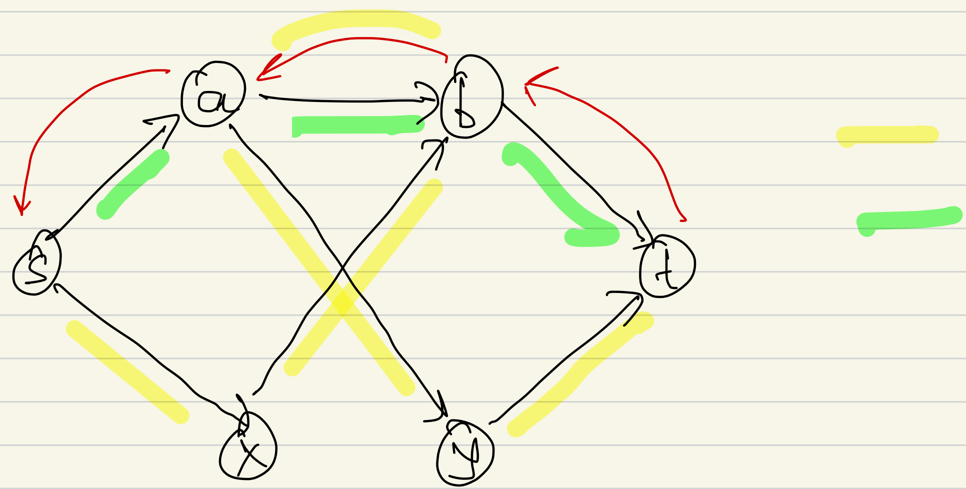
We can augment the flow by 1 along this path.

Recall: $a \to b$ had flow 1 in f

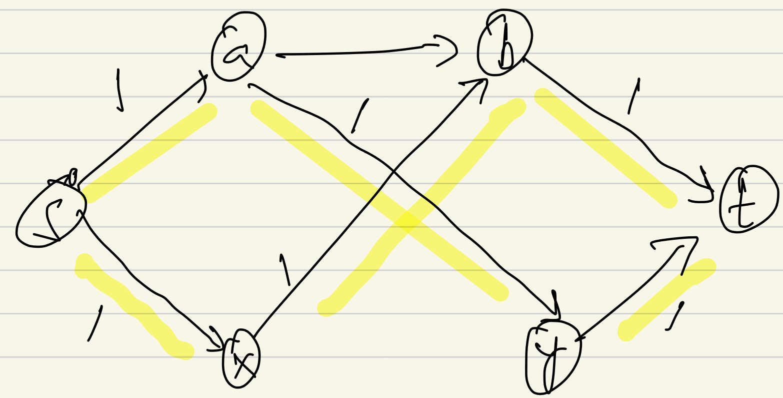
Now, we want to push 1 unit of flow from $b \to a$. $\Rightarrow$ Net flow from $a \to b = 0$.

(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, G, s, t, p)

$p \to$ $s \to t$ path in G_f

① $\delta = \min\limits_{e \in P} u_f(e)$

② For each $e \in P$,

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

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

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

③ Return f.

Properties of augmentation procedure

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

Lemma: f' is a valid flow.

Pf: Capacity constraint + conservation constraint

- Only changes in edges on the path p.

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

② $e = (x,y) \in E_{rev}$ (i.e. $(y,x) \in E$) by design $u_f(e) = f(x,y)$ and $\delta \le u_f(e)$.

So, $0 \le f'(e) \le u(e) \quad \forall e$.

Conservation constraints? — — — —

$\square$

Lemma 2

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

pf:

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

Look at the edge of p incident to t.

e

a) $e \in \vec{E}$

(b) $e \in E_{rev}$.

Ford-Fulkerson Algorithm

1) Initialize $f(e) = 0 \quad \forall e$

2) while true, do:

   (a) Construct G_f, u_f

   (b) If there is an s-t path p in G_f s.t. $\min_{e \in p} u_f > 0$,

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

   (c) Else Return f.

Correctness of Ford-Fulkerson

Lemma 1 If $u(e)$ is an integer for every $e \in E$

then,

1) FF terminates in $O(n \cdot m \cdot U)$ time
where $U = \max_{e \in E} u(e)$

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

Proof:

- We start with an integer valued flow. ($f(e) = 0 \; \forall e$)
- 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.

$\boxtimes$

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) $f(e) = u(e) \quad \forall\, e \in \partial^{+} S$

b) $f(e) = 0 \quad \forall\, e \in \partial^{-} S$

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

<u>Proof of Lemma 2</u>

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

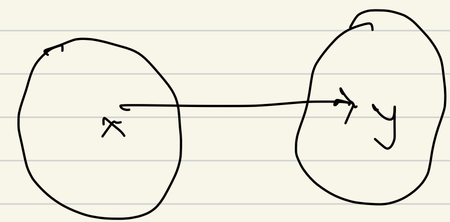
$S = \{ v$: v is reachable from s in G_f with all zero capacity edges removed.

1) $s \in S$ } why?

2) $t \notin S$ }

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

① Let $(x,y) \in \partial^+ S$

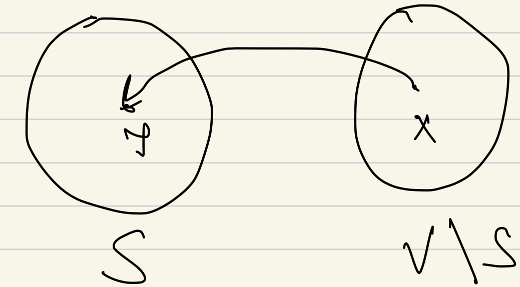


Then, $u_f(x,y) = 0$ } why?

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

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 → (also contains reversed edges)

x is reachable from s.

contradiction.

$\square$

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.