Max-flow:

Sending supply (e.g., internet traffic, products) from vertex s to vertex t without exceeding edge capacities.

Flow network:

directed graph $G=(V,E)$ with designated source $s \in V$ & sink $t \in V$.

for each $e \in E$, capacity $c_e > 0$.

Want to maximize flow from s to t.

f_e = flow along edge e.

Constraints:

1) for all $e \in E$, $0 \leq f_e \leq c_e$
2) for all $v \in V \setminus \{s \cup t\}$,

flow-in to v = flow-out of v

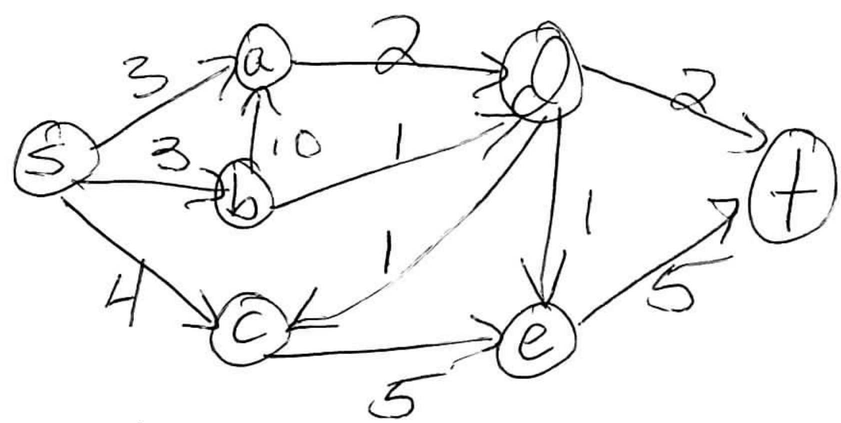
$$\sum_{\vec{wv} \in E} f_{wv} = \sum_{\vec{vz} \in E} f_{vz}$$

Satisfy capacity constraints

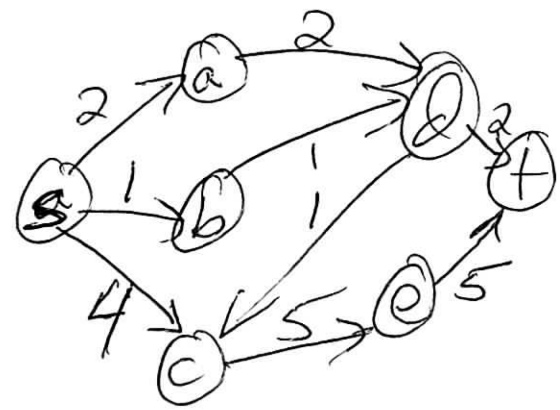
flow is conserved.

$$\text{size}(f) = \text{total flow sent}$$
$$= \text{flow-out of } s = \sum_{\vec{sv} \in E} f_{sv}$$
$$= \text{flow-in to } t = \sum_{\vec{zt} \in E} f_{zt}$$

Example from [DPV]:



flow network with edge capacities



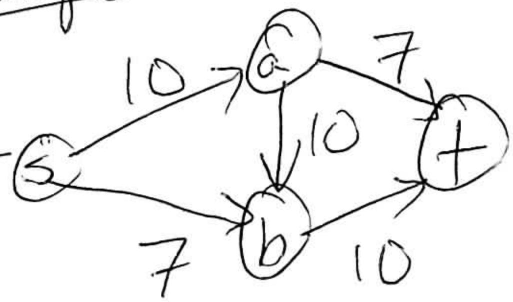
sample flow

Simple algorithm idea:

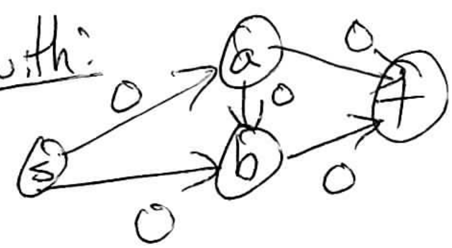
- Start with $f_e = 0$ for all $e \in E$
- find an s-t path P with available capacity
  - increase flow along P until some edge hits its capacity
  - repeat until no such s-t path.

Example:

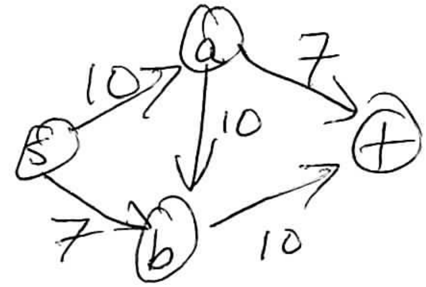
input:



start with:

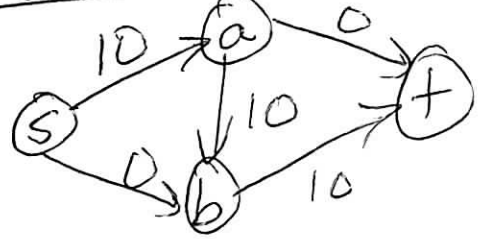


available capacities

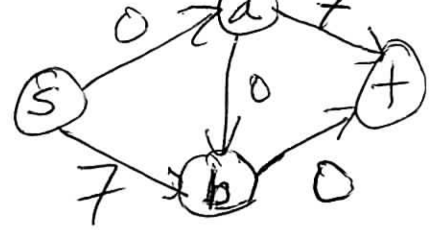


take path $s \to a \to b \to t$ & send 10 units along it.

new flow:



available capacities:



But no st-path here

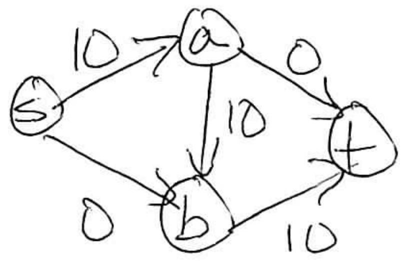
& the max-flow has value 17.

How do we increase further?

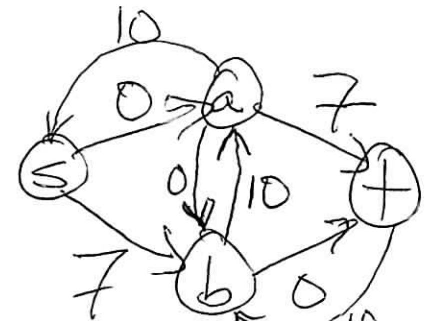
Send flow "backwards" along edge $a \to b$ by decreasing the flow along it.

So add edge $b \to a$ with available capacity 10.

current flow:

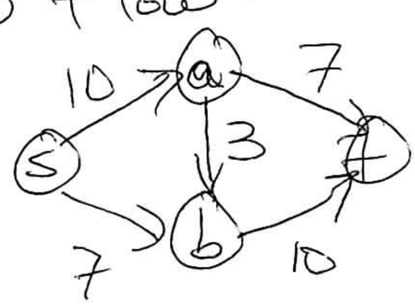


residual network:

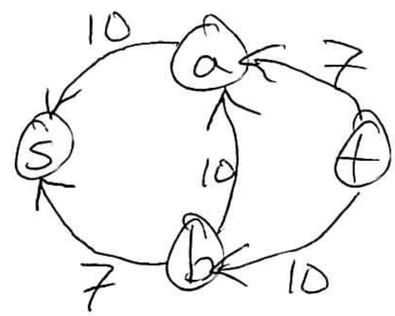


take Path $s \to b \to a \to t$ & send 7 units

new flow:



residual network:



No st-path in the residual network, hence we have a max-flow.

Formal definition of residual network:

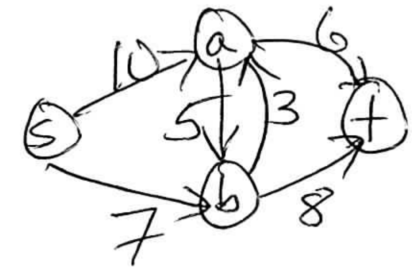
For flow network on $G=(V,E)$ with capacities c_e, & flow f_e, residual network $G^f=(V,E^f)$ has edges:

if $\overrightarrow{vw} \in E$ & $f_{vw} < c_{vw}$ then add $\overrightarrow{vw}$ to G^f with capacity $c_{vw} - f_{vw}$

if $vw \in E$ & $f_{vw} > 0$ then add $\overrightarrow{wv}$ to G^f with capacity f_{vw}

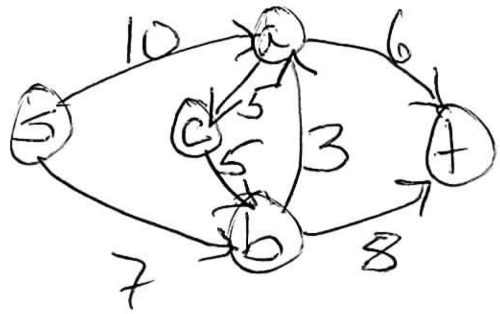
What happens if G has antiparallel edges?

Example:



How is the residual network defined in this case?

Easy solution: Remove antiparallel edges by using this equivalent graph:



So if edges $\overrightarrow{vw}$ & $\overrightarrow{wv}$
then add a new vertex z
& replace $\overrightarrow{vw}$ by $\overrightarrow{vz}, \overrightarrow{zw}$.

We now have an algorithm for solving max-flow.

Ford-Fulkerson algorithm:

1) Set $f_e = 0$ for all $e \in E$
2) Build the residual network G^f for the current flow f.
3) Check for a st-path P in G^f.

   if there is no such path then we are done & return (f) as the output.
4) Given P, let $c(P)$ = minimum capacity along P in G^f

   = "capacity" of P
5) Augment f by $c(P)$ units along P by:

   for a forward edge $e \in P$,
   increase f_e by $c(P)$

   for a backward edge $e \in P$, $e = \overrightarrow{vw}$,
   Decrease $f_{e'}$ by $c(P)$
   where $e' = \overrightarrow{wv} = e^R$.
6) Repeat until no st-path in G^f.

Running time:

if capacities are integers,
then the flow increases by ≥ 1 unit per round

So $\leq C$ rounds where
C = size of max-flow

Each round involves a DFS or BFS, so takes $O(|V|+|E|)$ time
$= O(|E|)$ assuming G is connected.

So running time is

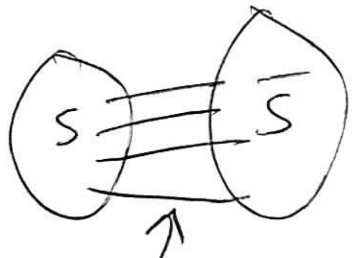
$$O(|E|C) = O(mC) \qquad m=|E| \quad n=|V|$$

$\Rightarrow$ Pseudo polynomial since it depends on the numbers in the input.

Better algorithm: we'll see [Edmonds-Karp '72]
takes the augmenting path in G^f which has fewest edges
then $O(nm)$ rounds
$\Rightarrow O(nm^2)$ total time.

For a graph $G=(V,E)$,

a <u>cut</u> is a partition $V = S \cup \bar{S}$.



edges of the cut.

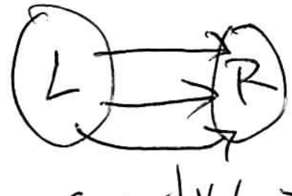
for directed $G=(V,E)$ with pair of vertices $s,t \in V$,

<u>st-cut</u> is a partition $V = L \cup R$

where $s \in L$, $t \in R$, $L \cap R = \emptyset$

Capacity of this st-cut is

$$\text{Capacity}(L,R) = \sum_{\substack{\vec{vw} \in E: \\ v \in L,\, w \in R}} c_{vw} = \begin{array}{c}\text{total}\\ \text{Capacity}\\ \text{from } L \to R\end{array}$$



Capacity $L \to R$.

for every flow f & every st-cut (L,R)

$$\text{size}(f) \leq \text{capacity}(L,R).$$

Why?

Because has to cross $L \to R$ &

$\leq$ capacity(L,R) units can cross it.

Max-flow min-cut theorem:

size of max-flow = min capacity of a st-cut

$$\max_f \text{size}(f) = \min_{(L,R)} \text{capacity}(L,R).$$

We'll show this theorem by proving that for a flow f, if there is no augmenting path in the residual network, then there is a st-cut (L,R) where $\text{size}(f) = \text{capacity}(L,R)$.

Hence, $\max_f \text{size}(f) \geq \min_{(L,R)} \text{capacity}(L,R)$.

& this shows that Ford-Fulkerson finds a max-flow.