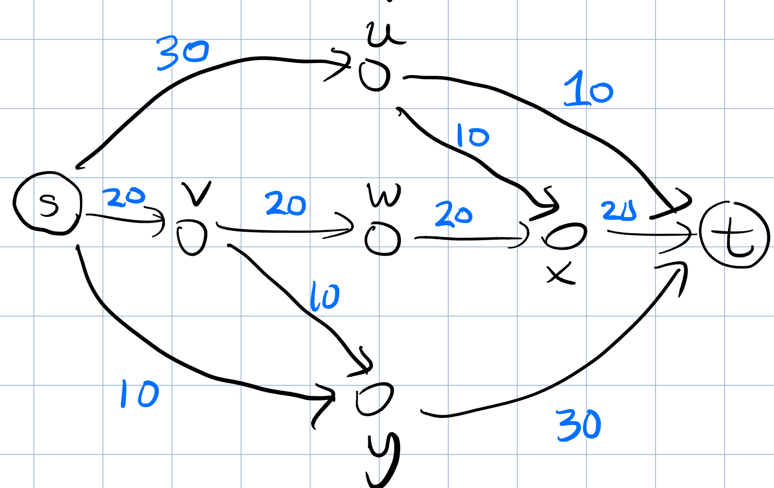
12 Mar 2021

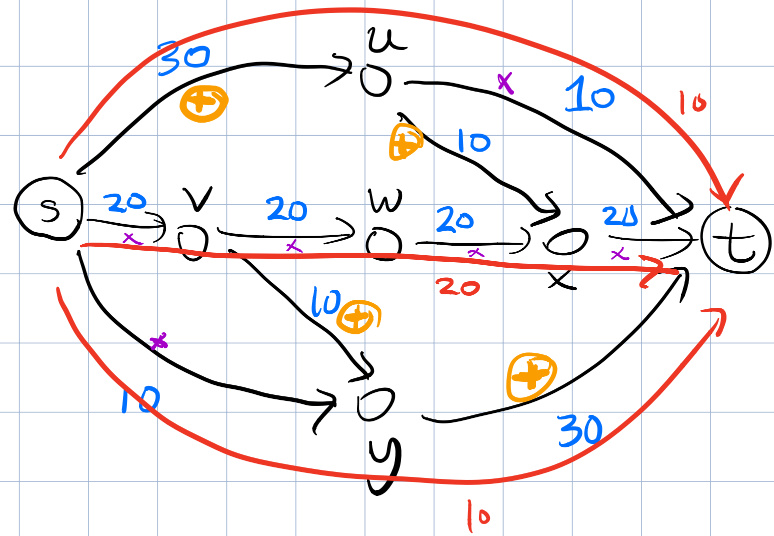
Network Flow and the Ford Fulkerson Algorithm (§7.1)

Announcement: Prelim 1 will be administered ONLINE in AT LEAST TWO TIME SLOTS. We will be running a poll to identify suitable time slots. There will be 15 minutes at the end to scan and upload your solutions.

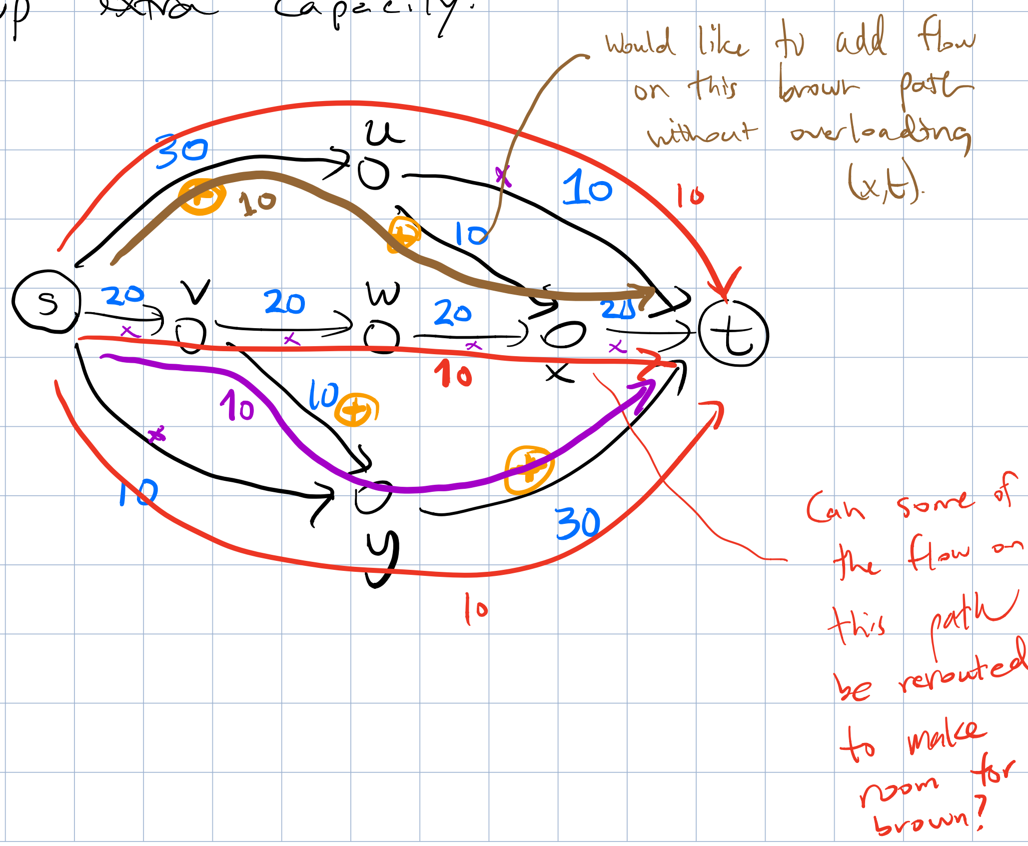
THEME of network flow: pack as many paths into a graph as possible.



Maximum # paths from s to t that we can pack into this graph without exceeding edge capacities?



At this point, to make further progress, we need to backtrack and reroute flow to free up extra capacity.



Formalizing the problem.

A <u>flow network</u> is a directed graph G with:

- vertex set V having two special vertices s (source), t (sink)
- edge set E having capacities $c(e) > 0$. (for this lecture $c(e)$ will be integers).
- every vertex belongs to at least one edge
- s has no incoming edges, t no outgoing.

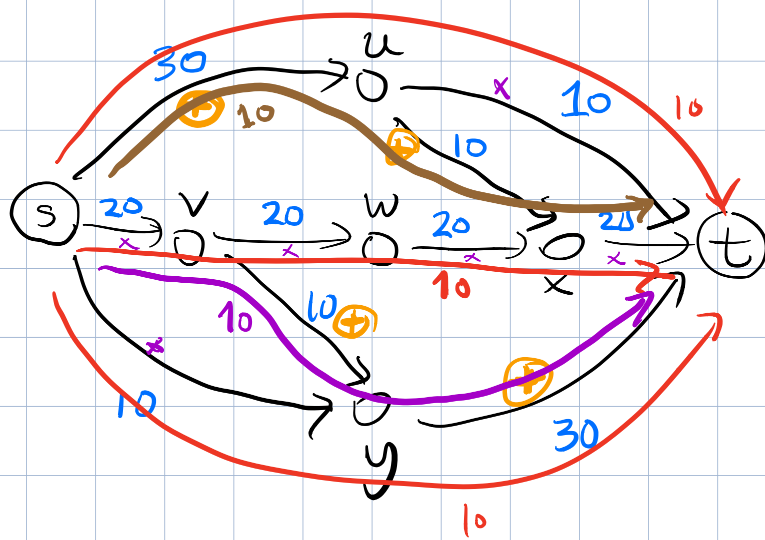
A <u>flow</u> in a flow network is an assignment of a number $f(e)$ to each edge e satisfying

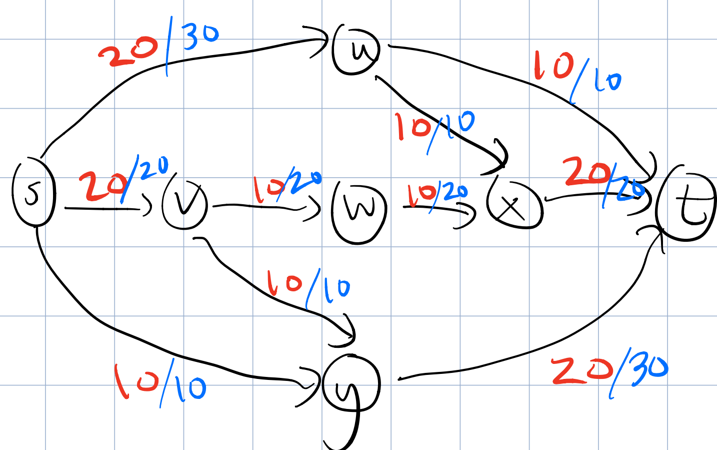
- [flow conservation] if $v \notin \{s,t\}$ then

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ from } v} f(e)$$

- [capacity constraints] $0 \le f(e) \le c(e)$ $\forall e$.

Notion: if we pack paths from s to t into G then $f(e)$ counts # paths using e.

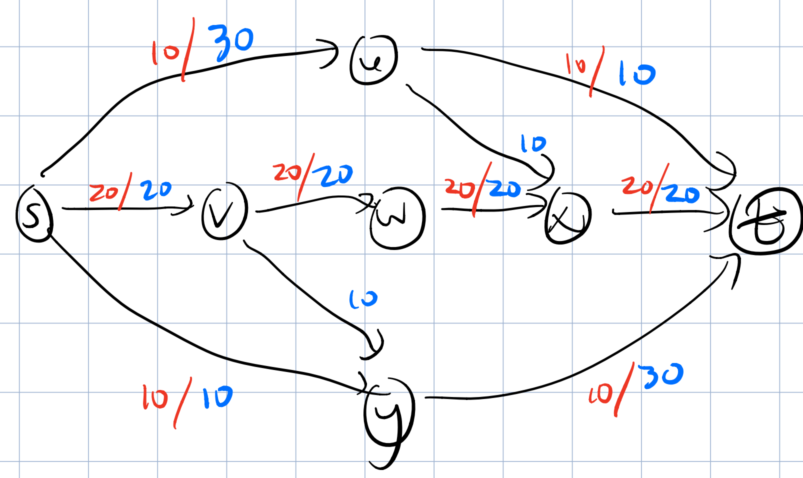




<u>Def.</u> The <u>value</u> of a flow is the quantity

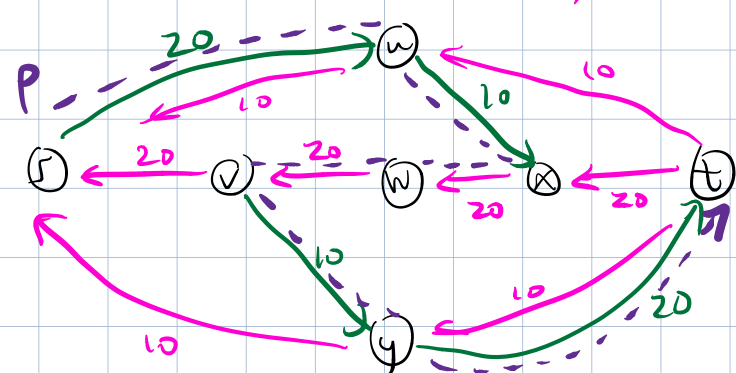
$$v(f) = \sum_{e \text{ out of } s} f(e)$$

The maximum flow problem is:
given flow network G, s, t, c
find a flow of maximum value.



Where is the unused capacity?

Where can we free up capacity by removing flow? [signify using reversed edges]



THE RESIDUAL GRAPH G_f

In general G_f has two types of edges:

- forward edge (u,v) if $e=(u,v) \in E(G)$ satisfies $f(e) < c(e)$ residual capacity $c(e) - f(e)$
- backward edge (u,v) if $e=(v,u) \in E(G)$ satisfies $f(e) > 0$ residual cap $f(e)$

Augmenting a path P

If G is a flow network
f is a flow
G_f is the residual graph
P is an s-t path in G_f

P is called an **augmenting path** for f.

AUGMENT(G, f, P): modify f using P to increase flow value.

let $b(f,P) = \min\{\text{residual cap of } e \mid e \in P\}$
"bottleneck value of P"

for each edge $(u,v) \in P$:
  if $e = (u,v)$ is a forward edge:
    $f'(e) = f(e) + b(f,P)$
  if (u,v) is a bkward edge:
    let $e = (v,u)$
    $f'(e) = f(e) - b(f,P)$

for each edge e not in P:
    $f'(e) = f(e)$

return f'.

This procedure modifies f to a new flow f' whose value is

$$v(f') = v(f) + b(f,P).$$