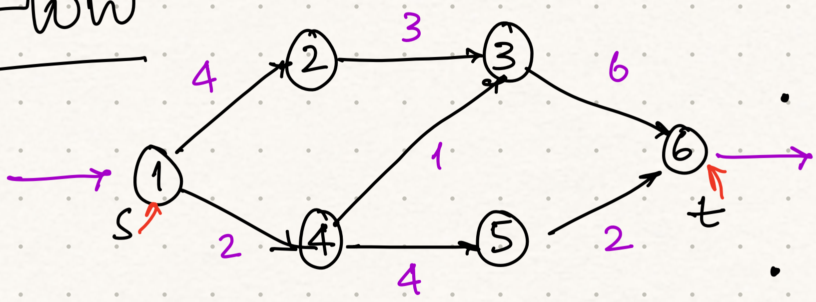
Network Flows

Max Flow



- Directed s-t graphs
- Capacities on edges $c(u \to v)$
- Flow reaching a vertex = Flow leaving a vertex. (Conservation of flow)

Obj: Maximize the flow between s to t.

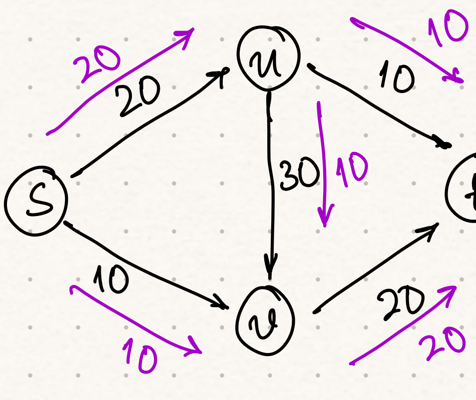
$$\left.\begin{array}{ll} \text{Capacities:} & c(u \to v) \\ \text{Flow:} & f(u \to v) \end{array}\right\} \forall\ u \to v \in E$$

1. $0 \le f(u \to v) \le c(u \to v)$

2. $\forall\ v \in V \setminus \{s,t\}$.

$$\underbrace{\sum_{\substack{u \in V \\ u \to v \in E}} f(u \to v)}_{\text{Incoming flow to } v} = \underbrace{\sum_{\substack{w \in V \\ v \to w \in E}} f(v \to w)}_{\text{Outgoing flow from } v.}$$

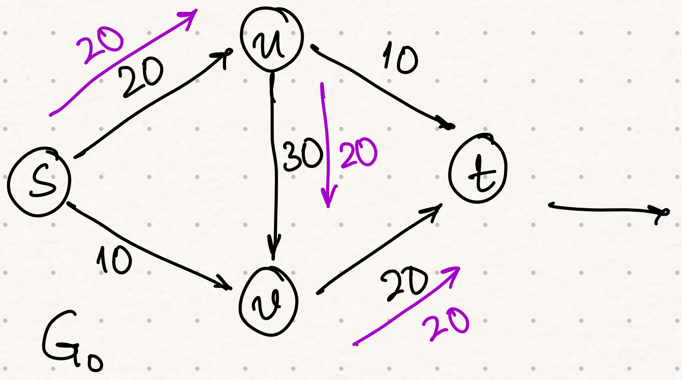
3. Maximize $\displaystyle\sum_{\substack{w \in V \\ s \to w \in E}} f(s \to w) \Big/ \sum_{\substack{u \in V \\ u \to t \in E}} f(u \to t)$

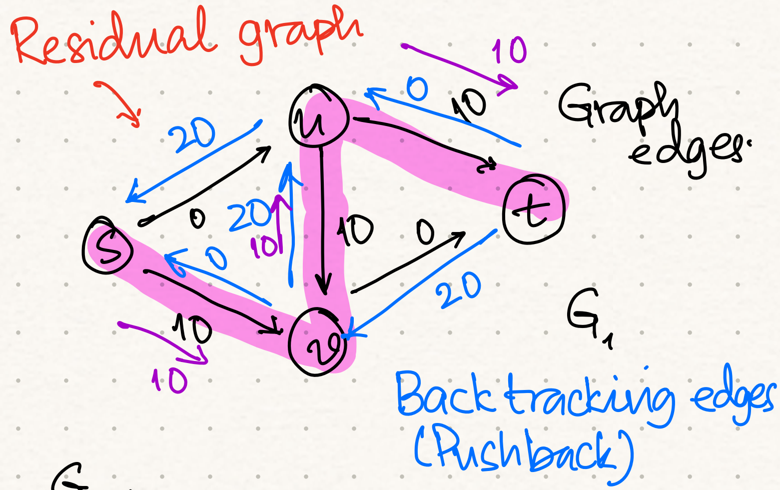


Max flow = 30.

1. Look for a $s \rightsquigarrow t$ path (BFS/DFS)

Obs: Look for a bottle neck.





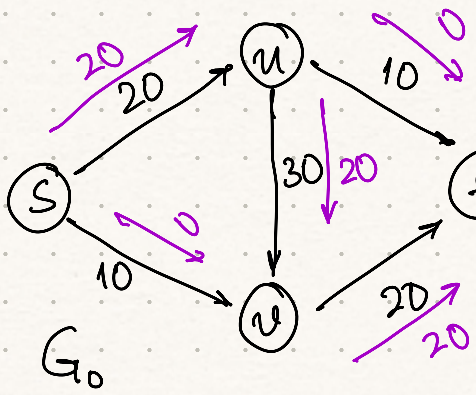
Look for ~~any~~ $s \rightsquigarrow t$ path in G_1.

$s \to v \to u \to t$ path has edges with >0 capacities.

Residual capacity (after flow f)

$$c_f(x \to y) = \begin{cases} c(x \to y) - f(x \to y) & \text{if } x \to y \in E \\ f(y \to x) & \text{if } y \to x \in E \\ 0 & \text{o/w} \end{cases}$$

for any $x, y \in V$
$x \neq y$



$c_f(s \to v) = 10 - 0 = 10$
$c_f(v \to s) = 0$

$c_f(s \to u) = 20 - 20 = 0$

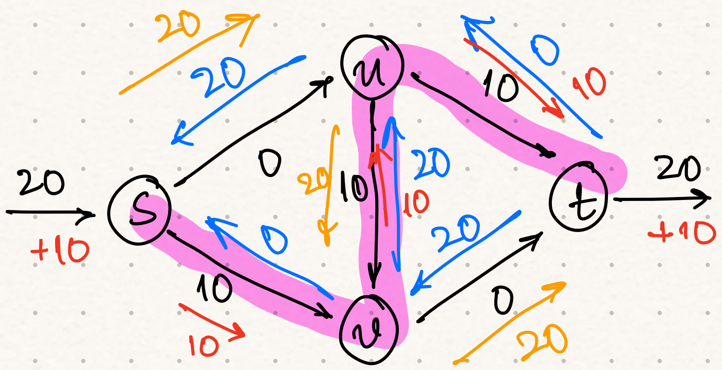
$c_f(u \to s) = 20 = f(s \to u)$ $\quad (s \to u \in E)$

$c_f(u \to v) = 30 - 20 = 10$

$c_f(v \to u) = f(u \to v) = 20$

$c_f(v \to t) = 0$

$c_f(t \to v) = f(v \to t) = 20$



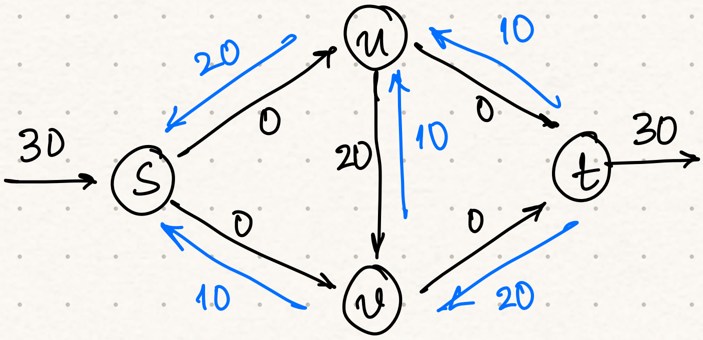
$s \to v \to u \to t$ is a path w/ >0 bottlenecks.

$c_f(s\to v) = 0$ $\qquad$ $c_f(v\to u) = 20-10=10$

$c_f(v\to s) = 10$ $\qquad$ $c_f(u\to v) = \underbrace{10-(-10)}= 20$

$c_f(u\to t) = 0$

$c_f(t\to u) = 10$



Obs: No $s \rightsquigarrow t$ paths in the residual graph.

$F \leftarrow 0$ / cap / $f(u\to v)=0 \;\; \forall\, u\to v \in E(G_{orig})$

Algo(G_{orig}, $\bar{0}$)

Algo(G, f)

1. Find a s–t path in G. (call it P) | If none,
   $O(m+n)$
2. Find the bottleneck on P; $b(P)$
   $O(n)$
3. Increment the flow by $b(P)$
   $F = F + b(P)$

$O(m+n)$

$$f'(u\to v) = \begin{cases} f(u\to v)+b(P) & \text{if } u\to v \in E(G_{orig}) \text{ and } u\to v \in P(G) \\ f(u\to v)-b(P) & \text{if } v\to u \in P(G) \text{ and } u\to v \in E(G_{orig}) \\ f(u\to v) & \text{otherwise.} \end{cases}$$

$O(m+n)$

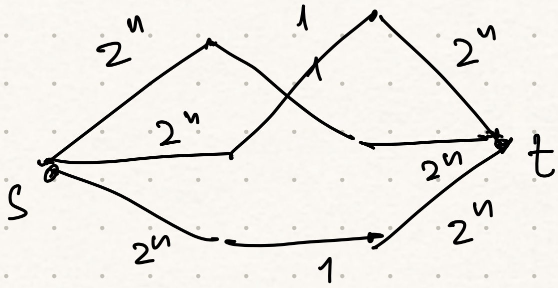
4. Compute the residual graph with f' values $\leftarrow G'$
5. Call for Algo(G', f').

Running time = # iterations × $O(m+n)$.

Obs: Flow increases in every step by ≥ 1 (if cap $\in \mathbb{Z}_{\geq 0}$)

iterations $\leq \Big(\sum c(s\to u)$

$$\min \left\{ \begin{array}{l} u: s\to u\in E \\ \displaystyle\sum_{v:\, v\to t\in E} c(v\to t) \end{array}\right.$$



Bound on iterations
$\leq 3\cdot 2^n$.

(Over