$\boxed{\text{8 Mar}}$ Ford-Fulkerson and Max-Flow Min-Cut

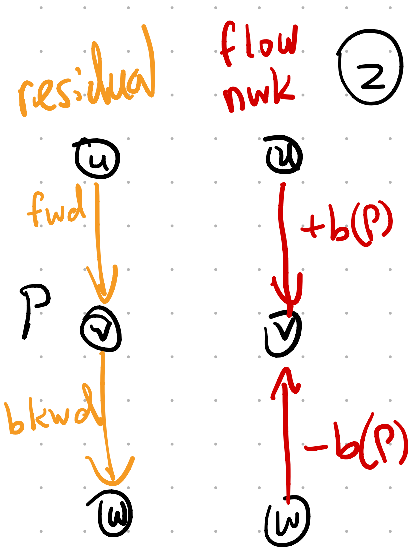
For a flow network (G, s, t, c) and a flow, f,

and augmenting path P (s-t in residual graph G_f)

define:

① "bottleneck capacity" $b(P) = \min_{e \in E(P)} \{ c_f(e) \}$

(minimum residual capacity on P)



② "augment f using P"

$$\forall e \in E(P) \begin{cases} f(e) \leftarrow f(e) + b(P) & \text{for } e \text{ forward edge} \\ f(\overleftarrow{e}) \leftarrow f(\overleftarrow{e}) - b(P) & \text{for } e \text{ backward edge} \end{cases}$$

When $e = (v, u)$, $\overleftarrow{e}$ denotes (u, v).

<u>Lemma:</u> ① If G is flow nwk, f is a flow, P is an augmenting path, and f' is the result of augmenting f using P, f' is a flow.

② If $c(e)$ and $f(e)$ are integer valued $\forall e$, then $f'(e)$ is also integer valued $\forall e$.

Ford-Fulkerson

Initialize $f(e) = 0$ for all edges E.

Initialize $G_f = G$.

while G_f contains an s-t path, P:

  $f \leftarrow$ augment f using P

  recalculate G_f

Output f.

Claim: If we run Ford-Fulkerson on an integer-capacitated network whose max flow has value V, it will terminate after at most $O(mV)$ steps.

\# edges (m) · max flow value (V)

Proof. $v(f)$ increases by $b(P) \geq 1$ in each iteration. Starts at 0, increases to $\leq V$, so while-loop iterates $\leq V$ times.

One while-loop iteration must:

* Search G_f for s-t path. BFS $O(m+n) = O(m)$.
* Find $b(P)$ $O(n) = O(m)$
* Augment f $O(n) = O(m)$
* Recalculate G_f $O(n) = O(m)$

$O(m)$ work per iteration

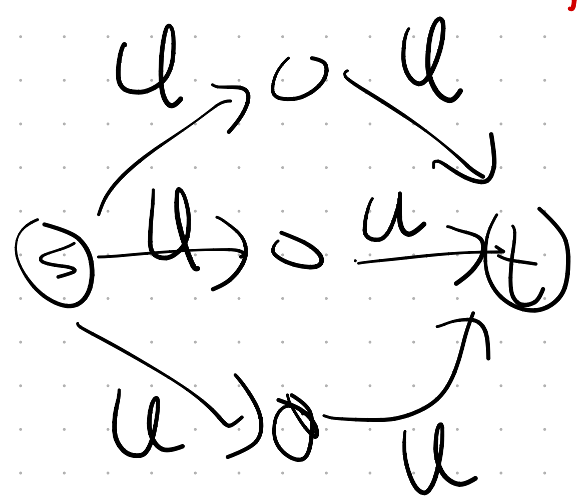
$\Longrightarrow$ $\boxed{O(mV)}$ running time.

"pseudopolynomial"

$\leq O(mC)$ where $C = \sum_{e \text{ out of } s} c(e)$

If edge capacities are in $\{1, \ldots, U\}$

then it takes $\log_2(U)+1$ bits to write an edge capacity, so input size is $O(m \log U)$ whereas V could be as large as $\frac{n \cdot U}{2}$.



V could be exponentially bigger than the input size.

Correctness?

Lemma says augmenting f using P preserves the property "f is a flow."
When algo terminates, it outputs a flow. Why maximum flow?

We will certify by finding a corresponding minimum cut.

Def. A cut in a flow network is a partition of the vertex set into A, B such that $s \in A$, $t \in B$.

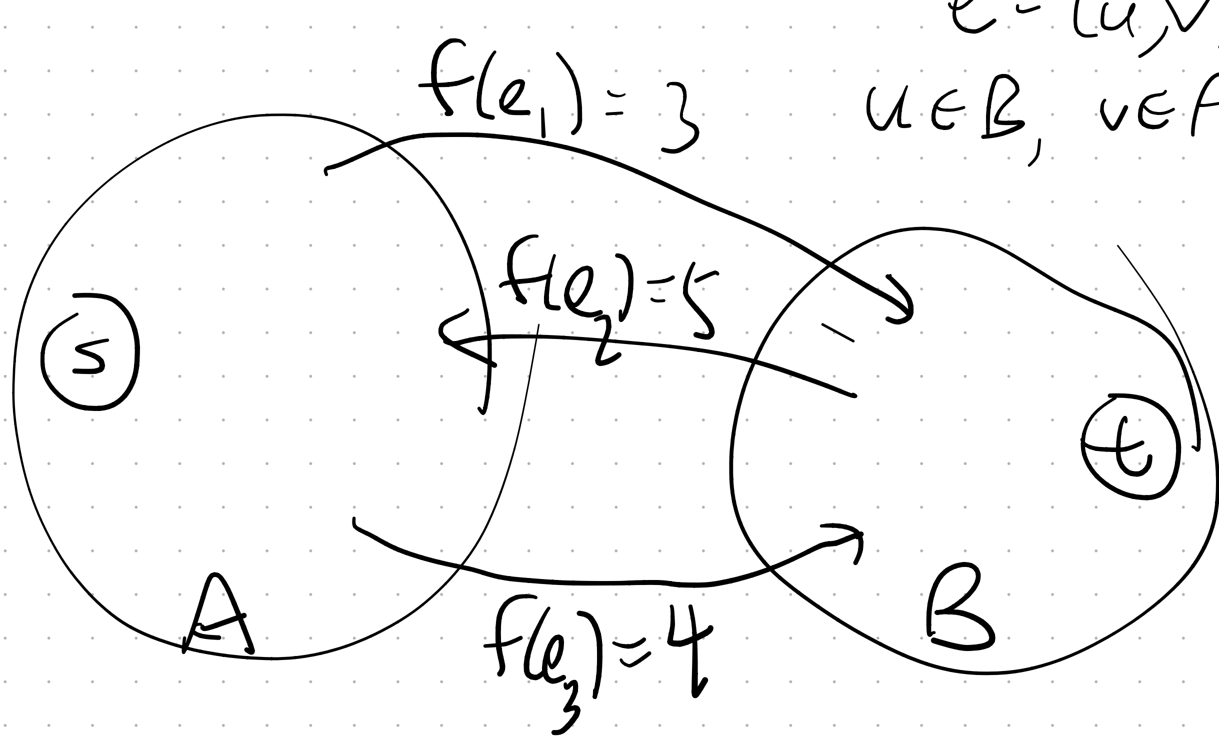
$A \cap B = \emptyset$
$A \cup B = V(G)$

The capacity of a cut is

$$c(A,B) = \sum_{\substack{e=(u,v) \\ u \in A,\, v \in B}} c(e).$$

Define $$f^{out}(A) = \sum_{\substack{e=(u,v) \\ u \in A,\, v \in B}} f(e)$$

$$f^{in}(A) = \sum_{\substack{e=(u,v) \\ u \in B,\, v \in A}} f(e).$$



$f^{out}(A) = 7$

$f^{in}(A) = 5.$

<u>Obs. 1.</u> If A, B is an s-t cut and f is a flow

(*) $$\underbrace{f^{out}(A) - f^{in}(A)}_{\sum_{v \in A} f^{out}(v) - f^{in}(v)} = v(f).$$

flow cons

<u>Obs 2.</u> $$f^{out}(A) \leq c(A, B)$$

$$f^{in}(A) \geq 0$$

$$v(f) = f^{out}(A) - f^{in}(A) \leq c(A,B)$$

value of <u>every</u> <u>flow</u>

$\leq$ capacity of <u>every</u> <u>cut</u>

$\Rightarrow$ $v(\text{max flow}) \leq \text{cap}(\text{min cut})$.

Max Flow Min Cut Theorem:

$$v(\text{max flow}) = \text{cap}(\text{min cut})$$