Given graph $G = (V, E)$ with edge capacities $c : E \rightarrow \mathbb{R} \geq 0$.

Theorem: Value of max flow $= \text{capacity of min cut}$.
Start with any feasible flow \( f: E \rightarrow \mathbb{R} \geq 0 \).

\[
\begin{align*}
\text{Residual capacities} \quad & C_f(n \rightarrow v) = \begin{cases} 
-c(n \rightarrow v) & \text{if } v \rightleftharpoons e \\
\delta(v \rightarrow w) & \text{if } v \rightarrow w \in E \\
0 & \text{o.w.}
\end{cases}
\end{align*}
\]
Residual graph $G_f = (V, E_f)$

of those with $\neq 0$ residual capacity

If $G_f$ has a path $P$ from $s$ to $t$, create a new flow $f'$ by "pushing" flow along $P$. 
Push \( \min \) residual capacity for edges on \( p \).

Value of flow increases by that much.

\[ \Rightarrow f \text{ was not maximum} \]

If no path from \( s \) to \( t \).

Let \( S \) be vertices in \( G \) you can reach from \( s \).

Let \( T = V \setminus S \).
For all $u \in S$, $v \in T$. If $u \nrightarrow v$ in $E$, then

$O = c_\sigma(u \nrightarrow v) = c(u \nrightarrow v) - f(u \nrightarrow v)$

it's saturated.

If $v \nrightarrow u$ in $E$, then

$O = c_\sigma(u \nrightarrow v) = f(v \nrightarrow v)$

it's avoided.
So $f$ is max value + ($S, T$) is min capacity with $|\mathcal{E}| = ||S_T||$. 
Ford-Fulkerson Augmenting Path Algorithm:

Start with an all-zero flow $\mathbf{f}$, $(\mathbf{f}(e)=0 \; \forall e \in \mathcal{E})$.

Repeatedly build residual graph and push flow along an augmenting path.

When you can't anymore, flow is maximum.
Analysis with integer capacities.

- Initial flow is all integers (0)

- If we assume each flow is all integers, residual capacities are integers

- So we always push a positive integer amount of flow

- So new flow is all integers
We push \( \geq 1 \) unit of flow each iteration.

Let \( f^* \) be the max value flow.

There are \( \leq 1f^* \) iterations.

O(E) time per iteration (build res. graph + search).

So, \( O(E1f^*) \) time total.
If \( f \) may not be polynomial in input size, may be exponential in \( A \) bits given as input.

Could do \( X \) iterations if you always push on \( u \rightarrow v \) or \( v \rightarrow u \).
"pseudo-polynomial time" - polynomial in terms of input numbers but not the input size (\# bits)

If \( |S| \) could be small depending on application
Algorithm may never terminate if given real-valued capacities.
Could imply infinite loops if from rounding floating point values!
Edmonds - Karp: “Fat pipes”

Choose the augmenting path with largest bottleneck capacity (maximize amount pushed)

\( O(E \log V) \) per iteration

Amount you have left to push decreases by a \( \frac{1}{E} \) fraction each iteration.
\[ \Rightarrow 1E_1 \cdot \ln |V|^\circ \text{ iterations} \]

(with integer capacities)

\[ O(E^2 \log V \log |V|^\circ) \]
Edmonds - Karp 2:
Choose augmenting path with smallest # edges.
O(E) time per iteration (use BFS in G_f)
Let $f$ be the flow after $i$ iterations.

Let $G = G_i$.

So $G = G_0$.

Let level $(v)$ be unweighted shortest path distance to $v$ in $G_i$. 
Lemma: \( \text{level}_u(v) \geq \text{level}_{\bar{u}}(v) \)

For all \( v \neq u \).

Proof: \( \text{level}_u(s) = 0 = \text{level}_{\bar{u}-1}(s) \) if.

Otherwise, let \( s \rightarrow \cdots \rightarrow u \rightarrow v \) be shortest path to \( v \) in \( G_{\bar{u}} \).

\( \text{level}_u(v) = \text{level}_{\bar{u}}(u) + 1 \).

By induction on \( \text{level}_{\bar{u}} \),

\( \text{level}_u(u) \geq \text{level}_{\bar{u}-1}(u) \).
If $u \rightarrow v$ is in $G$,
\[
\text{level}_{\hat{\omega}^{-1}}(u) + 1 \geq \text{level}_{\hat{\omega}^{-1}}(v)
\]

Otherwise, we must have push along $v \rightarrow u$. So shortest path to $u$ used $v \rightarrow u$, so
\[
\text{level}_{\hat{\omega}^{-1}}(u) + 1 > \text{level}_{\hat{\omega}^{-1}}(u) - 1 = \text{level}_{\hat{\omega}^{-1}}(v)
\]

Either way,
\[
\text{level}_{\hat{\omega}}(v) = \text{level}_{\hat{\omega}}(u) + 1 \\
\geq \text{level}_{\hat{\omega}^{-1}}(u) + 1 \\
\geq \text{level}_{\hat{\omega}^{-1}}(v)
\]
Lemma: Any edge \( u \rightarrow v \) disappears from residual graph \( E \) \( \leq |V| / 2 \) times.

Proof: Suppose \( u \rightarrow v \) in \( G_i \), but not in \( G_{i+1} \) but not in \( G_{i+1} \ldots G_j \).

\( u \rightarrow v \) in \( i \)th augmenting path, so \( \text{level}_u (v) = \text{level}_u (u) + 1 \).

\( v \rightarrow u \) in \( j \)th augmenting path, so \( \text{level}_j (u) = \text{level}_j (v) + 1 \).
So

\[ \text{level}_j(u) = \text{level}_j(v) + 1 \]

\[ \geq \text{level}_j(v) + 1 \]

\[ = \text{level}_j(v) + 2 \]

So distance from \( s \) to \( u \) increased by 2.

Max path length is \( 1|V| - 1 \)

So \( u \leftrightarrow v \) leaves \( \leq |V|^{1/2} \times \) times.
So EILNIV1z itera
tions,
$O(E^2V)$ time
(any $\ge 0$ real capacities)
"strongly polynomial time"

Dinitz (70s): $O(VEV^2)$ with
almost same algorithm

Orlin (2012): $O(VE)$ time

cite this