CSE 202: Design and Analysis of Algorithms

Lecture 10

Instructor: Kamalika Chaudhuri
Announcements

• Midterm on Feb 14 in class

• Material: Greedy, Divide and Conquer, Dynamic Programming, Flows (including Capacity Scaling, but \textbf{not} including Preflow-push)
**The Max Flow Problem:** Given directed graph $G=(V,E)$, source $s$, sink $t$, edge capacities $c(e)$, find an s-t flow of maximum size.

An s-t flow is a function $f:E \rightarrow R$ such that:
- $0 \leq f(e) \leq c(e)$, for all edges $e$
- flow into node $v$ = flow out of node $v$, for all nodes $v$ except $s$ and $t$,

\[ \sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e) \]

Size of flow $f =$ Total flow out of $s =$ total flow into $t$

Size of $f = 3$
The Max Flow Problem: Given directed graph $G=(V,E)$, source $s$, sink $t$, edge capacities $c(e)$, find an $s$-$t$ flow of maximum size

$\begin{align*}
&\text{The Residual Graph: For a flow } f \\
&G_f = (V, E_f) \text{ where } E_f \subseteq E \cup E^R \\
&\text{For any } (u,v) \text{ in } E \text{ or } E^R, \text{ residual capacity: } \ c_f(u,v) = c(u,v) - f(u,v) + f(v,u) \\
&[\text{ignore edges with zero } c_f; \text{ don’t put them in } E_f]
\end{align*}$
**Last Class: Facts about Flows**

**The Max Flow Problem:** Given directed graph $G=(V,E)$, source $s$, sink $t$, edge capacities $c(e)$, find an $s$-$t$ flow of maximum size.

**The Residual Graph:** For a flow $f$

$G_f = (V, E_f)$ where $E_f \subseteq E \cup E^R$

For any $(u,v)$ in $E$ or $E^R$, **residual capacity**:

$$c_f(u,v) = c(u,v) - f(u,v) + f(v,u)$$

[ignore edges with zero $c_f$: don’t put them in $E_f$]

**Max Flow Min Cut Theorem:**

Size(Max-Flow) = Capacity(Min-Cut)
The Max Flow Problem: Given directed graph $G=(V,E)$, source $s$, sink $t$, edge capacities $c(e)$, find an $s$-$t$ flow of maximum size.

The Residual Graph: For a flow $f$

$G_f = (V, E_f)$ where $E_f \subseteq E \cup E^R$

For any $(u,v)$ in $E$ or $E^R$, **residual capacity:**

$c_f(u,v) = c(u,v) - f(u,v) + f(v,u)$

[ignore edges with zero $c_f$: don’t put them in $E_f$]

Max Flow Min Cut Theorem:

Size(Max-Flow) = Capacity(Min-Cut)

When is $f$ a max flow?

When $t$ is not reachable from $s$ in $G_f$
Last Class: Algorithms for Max-Flow

Recall: $n = \#\text{vertices}, \ m = \#\text{edges in } G$

- Ford-Fulkerson: Running Time = $O(m F_{\text{max}})$

- Other efficient Ford-Fulkerson Style Algorithms:
  - Edmonds-Karp: Running Time = $O(nm^2)$
  - Capacity Scaling: Running Time = $O(m^2 \log C_{\text{max}})$

- Preflow-Push
Preflows

**Preflow:** A function \( f : E \rightarrow \mathbb{R} \) is a preflow if:

1. **Capacity Constraints:** \( 0 \leq f(e) \leq c(e) \)
2. Instead of conservation constraints:

\[
\sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e) \geq 0
\]

**Excess** \( (v) = \sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e) \)

**Example**

\[
\begin{align*}
G & \quad & f \\
\text{excess} &= 1 \\
\end{align*}
\]
Preflow-Push: Two Operations

**Preflow:** A function \( f: E \rightarrow \mathbb{R} \) is a preflow if:
1. **Capacity Constraints:** \( 0 \leq f(e) \leq c(e) \)
2. Instead of conservation constraints:
   \[
   \sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e) \geq 0
   \]

**Excess(\( v \)) = \sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e)**

**Labeling** \( h \) assigns a non-negative integer label \( h(v) \) to all \( v \) in \( V \)

**Push(\( v, w \)):** Applies if \( \text{excess}(v) > 0 \), \( h(w) < h(v) \), \( (v, w) \) in \( E_f \)
   \[
   q = \min(\text{excess}(v), c_f(v,w))
   \]
   Add \( q \) to \( f(v, w) \)

**Relabel(\( v \)):** Applies if \( \text{excess}(v) > 0 \), for all \( w \) s.t \( (v, w) \) in \( E_f \), \( h(w) \geq h(v) \)
   Increase \( h(v) \) by 1
Pre-Flow Push: The Algorithm

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for all other \( v \)
Start with preflow \( f: \) \( f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), for all other edges \( e \)

While there is a node (other than \( t \)) with positive excess
   Pick a node \( v \) with \( \text{excess}(v) > 0 \)
   If there is an edge \( (v, w) \) in \( E_f \) such that \( \text{push}(v, w) \) can be applied
      \( \text{Push}(v, w) \)
   Else
      \( \text{Relabel}(v) \)

**Push**(\( v, w \)): Applies if \( \text{excess}(v) > 0, h(w) < h(v), (v, w) \) in \( E_f \)
   \( q = \min(\text{excess}(v), c_f(v, w)) \)
   Add \( q \) to \( f(v, w) \)

**Relabel**(\( v \)): Applies if \( \text{excess}(v) > 0 \), for all \( w \) s.t \( (v, w) \) in \( E_f \), \( h(w) \geq h(v) \)
   Increase \( h(v) \) by 1
Pre-Flow Push

• Algorithm
• Correctness
• Running Time Analysis
Correctness: Proof Outline

Three Steps:

- Compatibility: Show that the preflow $f$ and the labeling $h$ maintained by the algorithm always obeys a compatibility property

- If a flow $f$ is compatible with some labeling, then $f$ is a max-flow

- Preflow-push outputs a flow on termination
Correctness: Compatible Pre-Flows

**Prefab:** A function $f: E \rightarrow \mathbb{R}$ is a preflow if:
1. **Capacity Constraints:** $0 \leq f(e) \leq c(e)$
2. Instead of conservation constraints:
   \[
   \sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e) \geq 0
   \]

**Excess(v) =** \[
\sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e)
\]

Preflow $f$ and labeling $h$ are compatible if:
1. $h(s) = n, h(t) = 0$
2. For all edges $(v, w)$ in the residual graph $G_f$, $h(v) \leq h(w) + 1$

**Invariant:** Preflow $f$ and labeling $h$ are always compatible over the Preflow-Push algorithm
Running Time Analysis: Outline

1. How many Push Ops? Relabel Ops?
2. How to implement Push and Relabel Ops efficiently?
Running Time Analysis: Outline

1. How many Relabel Ops?

**Main Idea:** Bound the maximum value of $h(v)$ for any node $v$, and bound #relabel ops through this
## Preflow Push: #Relabels

**Property 1:** In a preflow \( f \), if \( \text{excess}(v) > 0 \), then there is a path from \( v \) to \( s \) in \( G_f \)

- **Start with labeling:** \( h(s) = n, h(t) = 0, h(v) = 0, \) for other \( v \)
- **Start with preflow:** \( f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess
- Pick a node \( v \) with \( \text{excess}(v) > 0 \)
- If there is an edge \( (v, w) \) in \( E_f \) s.t. \( \text{push}(v, w) \) applies
  - \( \text{Push}(v, w) \)
- Else
  - \( \text{Relabel}(v) \)

- **Push(\( v, w \)):**
  - Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
  - \( q = \min(\text{excess}(v), cf(v,w)) \)
  - Add \( q \) to \( f(v, w) \)

- **Relabel(\( v \)):**
  - Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t \( (v, w) \) in \( E_f \), \( h(w) >= h(v) \)
  - Increase \( h(v) \) by 1

```
\begin{tabular}{|c|c|}
<table>
<thead>
<tr>
<th>s</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>\text{A} = \text{all nodes } v \text{ s.t. } s \text{ is reachable from } v \text{ in } G_f</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>\text{B} = \text{remaining nodes}</td>
<td></td>
</tr>
</tbody>
</table>
\end{tabular}
```
Preflow Push: #Relabels

Start with labeling: $h(s) = n, h(t) = 0, h(v) = 0$, for other $v$
Start with preflow $f$: $f(e) = c(e)$ for $e = (s, v), f(e) = 0$, ow

While there is a node (other than $t$) with positive excess
- Pick a node $v$ with $excess(v) > 0$
  - If there is an edge $(v, w)$ in $G_f$ s.t. $push(v, w)$ applies
    - Push($v, w$)
  - Else
    - Relabel($v$)

Property 1: In a preflow $f$, if $excess(v) > 0$, then there is a path from $v$ to $s$ in $G_f$

$$s \xrightarrow{} x \xrightarrow{} y \xrightarrow{} t$$
A = all nodes $v$ s.t. $s$ is reachable from $v$ in $G_f$
B = remaining nodes

Fact: Any $e = (x, y)$ from $A$ to $B$ has $f(x, y) = 0$
If not, $(y, x)$ is in $G_f$, so there is a $y$ - $s$ path

Push($v$, $w$):
Applies if $excess(v) > 0$, $h(w) < h(v)$
$$q = min(excess(v), c_f(v, w))$$
Add $q$ to $f(v, w)$

Relabel($v$):
Applies if $excess(v) > 0$ and for all $w$ s.t. $(v, w)$ in $G_f$, $h(w) >= h(v)$
Increase $h(v)$ by 1
Preflow Push: #Relabels

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess
Pick a node \( v \) with \( \text{excess}(v) > 0 \)
If there is an edge \( (v, w) \) in \( G_f \) s.t. push\((v, w)\) applies
    Push\((v, w)\)
Else
    Relabel\((v)\)

**Property 1**: In a preflow \( f \), if \( \text{excess}(v) > 0 \), then there is a path from \( v \) to \( s \) in \( G_f \)

Now, total excess of nodes in \( B = \)
\[
\sum_{v \in B} \sum_{e \text{ into } B} f(e) - \sum_{v \in B} \sum_{e \text{ out of } B} f(e) \geq 0
\]

**Push\((v, w)\)**:
Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
    \( q = \min(\text{excess}(v), c_f(v,w)) \)
    Add \( q \) to \( f(v, w) \)

**Relabel\((v)\)**:
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t \( (v, w) \) in \( E_f \), \( h(w) \geq h(v) \)
    Increase \( h(v) \) by 1

**Fact**: Any \( e=(x, y) \) from \( A \) to \( B \) has \( f(x,y) = 0 \)
If not, \( (y, x) \) is in \( G_f \), so there is a \( y - s \) path
Preflow Push: #Relabels

Property 1: In a preflow $f$, if $\text{excess}(v) > 0$, then there is a path from $v$ to $s$ in $G_f$.

**Push($v$, $w$):**
Applies if $\text{excess}(v) > 0$, $h(w) < h(v)$
$q = \min(\text{excess}(v), c_f(v, w))$
Add $q$ to $f(v, w)$

**Relabel($v$):**
Applies if $\text{excess}(v) > 0$ and for all $w$ s.t. $(v, w)$ in $E_f$, $h(w) \geq h(v)$
Increase $h(v)$ by 1

Now, total excess of nodes in $B = \sum_{v \in B} \sum_{e \text{ into } B} f(e) - \sum_{v \in B} \sum_{e \text{ out of } B} f(e) \geq 0$

Three types of edges $e$ in the sum:

Property 1: In a preflow $f$, if $\text{excess}(v) > 0$, then there is a path from $v$ to $s$ in $G_f$.

Fact: Any $e=(x, y)$ from $A$ to $B$ has $f(x, y) = 0$
If not, $(y, x)$ is in $G_f$, so there is a $y$ - $s$ path
Preflow Push: #Relabels

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)

Start with preflow \( f \): \( f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess

Pick a node \( v \) with \( \text{excess}(v) > 0 \)

If there is an edge \((v, w)\) in \( E_f\) s.t. push(v, w) applies

Push(v, w)

Else

Relabel(v)

**Property 1:** In a preflow \( f \), if \( \text{excess}(v) > 0 \), then there is a path from \( v \) to \( s \) in \( G_f \)

\[
\sum_{v \in B} \sum_{e \text{ into } B} f(e) - \sum_{v \in B} \sum_{e \text{ out of } B} f(e) \geq 0
\]

Three types of edges \( e \) in the sum:

1. Both endpoints of \( e \) are in \( B \): \( f(e) \) cancels out

**Fact:** Any \( e=(x, y) \) from \( A \) to \( B \) has \( f(x, y) = 0 \)

If not, \((y, x)\) is in \( G_r \), so there is a \( y - s \) path
Preflow Push: #Relabels

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f \): \( f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess
   Pick a node \( v \) with \( \text{excess}(v) > 0 \)
   If there is an edge \( (v, w) \) in \( E_f \) s. t. \( \text{push}(v, w) \) applies
      Push\((v, w)\)
   Else
      Relabel\((v)\)

**Property 1:** In a preflow \( f \), if \( \text{excess}(v) > 0 \), then there is a path from \( v \) to \( s \) in \( G_f \)

\[
\begin{array}{c}
s \\
A \\
x \\
y \\
B \\
t
\end{array}
\]

A = all nodes \( v \) s.t. \( s \) is reachable from \( v \) in \( G_f \)
B = remaining nodes

**Fact:** Any \( e = (x, y) \) from A to B has \( f(x, y) = 0 \)
If not, \( (y, x) \) is in \( G_f \), so there is a y - s path

**Push\((v, w)\):**
Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
   \( q = \min(\text{excess}(v), c_f(v, w)) \)
   Add \( q \) to \( f(v, w) \)

**Relabel\((v)\):**
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t \( (v, w) \) in \( E_f \), \( h(w) \geq h(v) \)
   Increase \( h(v) \) by 1

Now, total excess of nodes in B =
\[
\sum_{v \in B} \left( \sum_{e \text{ into } B} f(e) - \sum_{v \in B} \sum_{e \text{ out of } B} f(e) \right) \geq 0
\]

Three types of edges \( e \) in the sum:
1. Both endpoints of \( e \) are in B: \( f(e) \) cancels out
2. \( e = (u, v) \), u in A, v in B: \( f(e) = 0 \)
Preflow Push: #ReLabels

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f \): \( f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess
   Pick a node \( v \) with \( \text{excess}(v) > 0 \)
   If there is an edge \((v, w)\) in \( E_f \) s. t. \( \text{push}(v, w) \) applies
      \( \text{Push}(v, w) \)
   Else
      \( \text{Relabel}(v) \)

Property 1: In a preflow \( f \), if \( \text{excess}(v) > 0 \), then there is a path from \( v \) to \( s \) in \( G_f \)

Push(\( v, w \)):
Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
\( q = \min(\text{excess}(v), c_f(v,w)) \)
Add \( q \) to \( f(v, w) \)

Relabel(\( v \)):
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s. t \((v, w)\) in \( E_f \), \( h(w) \geq h(v) \)
Increase \( h(v) \) by 1

Now, total excess of nodes in \( B \) =
\[
\sum_{v \in B} \sum_{e \text{ into } B} f(e) - \sum_{v \in B} \sum_{e \text{ out of } B} f(e) \geq 0
\]

Three types of edges \( e \) in the sum:
1. Both endpoints of \( e \) are in \( B \): \( f(e) \) cancels out
2. \( e = (u, v) \), \( u \) in \( A \), \( v \) in \( B \): \( f(e) = 0 \)
3. \( e = (v, u) \), \( u \) in \( A \), \( v \) in \( B \)

Fact: Any \( e = (x, y) \) from \( A \) to \( B \) has \( f(x,y) = 0 \)
If not, \( (y, x) \) is in \( G_f \), so there is a \( y - s \) path
Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess
Pick a node \( v \) with \( \text{excess}(v) > 0 \)
If there is an edge \( (v, w) \) in \( E_f \) s.t. \( \text{push}(v, w) \) applies
Push\( (v, w) \)
Else
Relabel\( (v) \)

**Property 1:** In a preflow \( f \), if \( \text{excess}(v) > 0 \), then there is a path from \( v \) to \( s \) in \( G_f \)

**Fact:** Any \( e = (x, y) \) from \( A \) to \( B \) has \( f(x, y) = 0 \)
If not, \( (y, x) \) is in \( G_f \), so there is a \( y - s \) path

**Push\( (v, w) \):**
Applies if \( \text{excess}(v) > 0 \), \( h(w) < h(v) \)
\[ q = \min(\text{excess}(v), c_f(v,w)) \]
Add \( q \) to \( f(v, w) \)

**Relabel\( (v) \):**
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t. \( (v, w) \) in \( E_f \), \( h(w) \geq h(v) \)
Increase \( h(v) \) by 1

Now, total excess of nodes in \( B = \)
\[
\sum_{v \in B} \left( \sum_{e \text{ into } B} f(e) - \sum_{e \text{ out of } B} f(e) \right) \geq 0
\]

Three types of edges \( e \) in the sum:
1. Both endpoints of \( e \) are in \( B \): \( f(e) \) cancels out
2. \( e = (u, v) \), \( u \) in \( A \), \( v \) in \( B \): \( f(e) = 0 \)
3. \( e = (v, u) \), \( u \) in \( A \), \( v \) in \( B \)

Total excess of nodes in \( B = - \sum_{v \in B} \sum_{u \in A} f(v, u) \geq 0 \)
Preflow Push: #Relabels

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess
Pick a node \( v \) with \( \text{excess}(v) > 0 \)
If there is an edge \( (v, w) \) in \( E_f \) s.t. \( \text{push}(v, w) \) applies
  Push \((v, w)\)
Else
  Relabel \((v)\)

**Property 1:** In a preflow \( f \), if \( \text{excess}(v) > 0 \), then there is a path from \( v \) to \( s \) in \( G_f \)

![Diagram](https://via.placeholder.com/150)

**A** = all nodes \( v \) s.t. \( s \) is reachable from \( v \) in \( G_f \)
**B** = remaining nodes

**Fact:** Any \( e = (x, y) \) from \( A \) to \( B \) has \( f(x, y) = 0 \)
If not, \( (y, x) \) is in \( G_f \), so there is a \( y - s \) path

**Push \((v, w)\):**
Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
  \( q = \min(\text{excess}(v), c_f(v, w)) \)
  Add \( q \) to \( f(v, w) \)

**Relabel \((v)\):**
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t \((v, w)\) in \( E_f \), \( h(w) \geq h(v) \)
  Increase \( h(v) \) by \( 1 \)

Now, total excess of nodes in \( B = \)
\[
\sum_{v \in B} \sum_{e \text{ into } B} f(e) - \sum_{v \in B} \sum_{e \text{ out of } B} f(e) \geq 0
\]

Three types of edges \( e \) in the sum:
1. Both endpoints of \( e \) are in \( B \): \( f(e) \) cancels out
2. \( e = (u, v) \), \( u \) in \( A \), \( v \) in \( B \): \( f(e) = 0 \)
3. \( e = (v, u) \), \( u \) in \( A \), \( v \) in \( B \)

Total excess of nodes in \( B = \)- \( \sum_{v \in B} \sum_{u \in A} f(v, u) \geq 0 \)

As \( \text{excess}(v) \) is never \(<0\), \( \text{excess}(v) = 0 \) for \( v \) in \( B \)
Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)

Start with preflow \( f \): \( f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess

Pick a node \( v \) with \( \text{excess}(v) > 0 \)

If there is an edge \( (v, w) \) in \( E_f \) s. t. \( \text{push}(v, w) \) applies

Push\((v, w)\)

Else

Relabel\((v)\)

**Push\((v, w)\):**
Applies if \( \text{excess}(v) > 0 \), \( h(w) < h(v) \)
\[ q = \min(\text{excess}(v), c_f(v, w)) \]
Add \( q \) to \( f(v, w) \)

**Relabel\((v)\):**
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s. t. \( (v, w) \) in \( E_f \), \( h(w) \geq h(v) \)
Increase \( h(v) \) by 1

**Compatibility** of \( f \) and \( h \):
1. \( h(s) = n, h(t) = 0 \)
2. For all edges \( (v, w) \) in \( G_f \),
   \[ h(v) \leq h(w) + 1 \]

**Property 1:** In a preflow \( f \), if \( \text{excess}(v) > 0 \), then there is a path from \( v \) to \( s \) in \( G_f \)

**Property 2:** At any point, for any \( v \), \( h(v) \leq 2n - 1 \)
Preflow Push: #Relabels

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f: f(e) = c(e) \) for \( e = (s,v), f(e) = 0, \) ow

While there is a node (other than \( t \)) with positive excess
Pick a node \( v \) with excess\((v) > 0 \)
If there is an edge \( (v, w) \) in \( E_f \) s.t. push\((v, w) \) applies
   Push\((v, w)\)
Else
   Relabel\((v)\)

### Property 1:
In a preflow \( f \), if excess\((v) > 0 \), then there is a path from \( v \) to \( s \) in \( G_f \)

### Property 2:
At any point, for any \( v \), \( h(v) \leq 2n - 1 \)

**Proof:** If excess\((v) > 0 \), there is a \( v-s \) path in \( G_f \)
Let \( v = v_1, \ldots, v_k = s \) be the path

---

**Push\((v, w)\):**
Applies if excess\((v) > 0, h(w) < h(v) \)
\( q = \min(\text{excess}(v), c_f(v,w)) \)
Add \( q \) to \( f(v, w) \)

**Relabel\((v)\):**
Applies if excess\((v) > 0 \) and for all \( w \) s.t \( (v, w) \) in \( E_f \), \( h(w) \geq h(v) \)
Increase \( h(v) \) by 1

**Compatibility** of \( f \) and \( h \):
1. \( h(s) = n, h(t) = 0 \)
2. For all edges \( (v, w) \) in \( G_f \), \( h(w) \geq h(v) \)
   \( h(v) \leq h(w) + 1 \)
Preflow Push: \#Relabels

Start with labeling: $h(s) = n, h(t) = 0, h(v) = 0$, for other $v$
Start with preflow $f$: $f(e) = c(e)$ for $e = (s, v), f(e) = 0$, ow

While there is a node (other than $t$) with positive excess
Pick a node $v$ with $excess(v) > 0$
If there is an edge $(v, w)$ in $G_f$ s. t. push($v, w$) applies
   Push($v, w$)
Else
   Relabel($v$)

Property 1: In a preflow $f$, if $excess(v) > 0$, then there is a path from $v$ to $s$ in $G_f$

Property 2: At any point, for any $v$, $h(v) \leq 2n - 1$

Proof: If $excess(v) > 0$, there is a $v$-$s$ path in $G_f$
Let $v = v_1, ..., v_k = s$ be the path
By compatibility:
$h(s) = n, h(v_{k-1}) \leq n + 1, h(v_1) \leq n+k-1 \leq 2n - 1$

Push($v, w$):
Applies if $excess(v) > 0$, $h(w) < h(v)$
   $q = \min(excess(v), c_f(v,w))$
   Add $q$ to $f(v, w)$

Relabel($v$):
Applies if $excess(v) > 0$ and for all $w$ s. t. $(v, w)$ in $E_f$, $h(w) \geq h(v)$
   Increase $h(v)$ by 1

Compatibility of $f$ and $h$:
1. $h(s) = n, h(t) = 0$
2. For all edges $(v, w)$ in $G_f$, $h(v) \leq h(w) + 1$
Preflow Push: #Relabels

Start with labeling: $h(s) = n, h(t) = 0, h(v) = 0$, for other $v$
Start with preflow $f$: $f(e) = c(e)$ for $e = (s, v), f(e) = 0$, ow

While there is a node (other than $t$) with positive excess
Pick a node $v$ with $\text{excess}(v) > 0$
If there is an edge $(v, w)$ in $G_f$ s.t. push$(v, w)$ applies
   Push$(v, w)$
Else
   Relabel$(v)$

Property 1: In a preflow $f$, if $\text{excess}(v) > 0$, then there is a path from $v$ to $s$ in $G_f$

Property 2: At any point, for any $v$, $h(v) \leq 2n - 1$

Proof: If $\text{excess}(v) > 0$, there is a $v$-s path in $G_f$
Let $v = v_1, ..., v_k = s$ be the path
By compatibility:
$h(s) = n, h(v_{k-1}) \leq n + 1, h(v_1) \leq n+k-1 \leq 2n - 1$

Push$(v, w)$:
Applies if $\text{excess}(v) > 0, h(w) < h(v)$
$q = \min(\text{excess}(v), c_r(v, w))$
Add $q$ to $f(v, w)$

Relabel$(v)$:
Applies if $\text{excess}(v) > 0$ and for all $w$ s.t $(v, w)$ in $G_f$, $h(w) \geq h(v)$
Increase $h(v)$ by 1

Compatibility of $f$ and $h$:
1. $h(s) = n, h(t) = 0$
2. For all edges $(v, w)$ in $G_f$, $h(v) \leq h(w) + 1$

If $\text{excess}(v) = 0$, then $h(v)$ has not changed since the last time $v$ had excess $> 0$
Thus, $h(v) \leq 2n - 1$ also

\[ 
\begin{matrix}
\text{v1} & \text{v2} & \text{v3} & \ldots & \text{vk-1} & \text{s = vk} \\
\end{matrix}
\]
Preflow Push: #Relabels

Property 1: In a preflow $f$, if $\text{excess}(v) > 0$, then there is a path from $v$ to $s$ in $G_f$.

Property 2: At any point in the algorithm, for any $v$, $h(v) \leq 2n - 1$.

Start with labeling: $h(s) = n, h(t) = 0, h(v) = 0$, for other $v$.

Start with preflow $f$: $f(e) = c(e)$ for $e = (s, v)$, $f(e) = 0$, otherwise.

While there is a node (other than $t$) with positive excess:
- Pick a node $v$ with $\text{excess}(v) > 0$.
- If there is an edge $(v, w)$ in $E_f$ such that $\text{push}(v, w)$ applies:
  - Push($v, w$)
- Else:
  - Relabel($v$)

Push($v, w$):
Applies if $\text{excess}(v) > 0$, $h(w) < h(v)$.
$\q = \min(\text{excess}(v), c_f(v, w))$
Add $\q$ to $f(v, w)$

Relabel($v$):
Applies if $\text{excess}(v) > 0$ and for all $w$ such that $(v, w) \in E_f$, $h(w) \geq h(v)$.
Increase $h(v)$ by 1.
**Preflow Push: #Relabels**

Start with labeling: \(h(s) = n, h(t) = 0, h(v) = 0\), for other \(v\)

Start with preflow \(f\): \(f(e) = c(e)\) for \(e = (s, v)\), \(f(e) = 0\), ow

While there is a node (other than \(t\)) with positive excess

- Pick a node \(v\) with \(\text{excess}(v) > 0\)
- If there is an edge \((v, w)\) in \(E_f\) s.t. \(\text{push}(v, w)\) applies
  - \(\text{Push}(v, w)\)
- Else
  - \(\text{Relabel}(v)\)

**Property 1:** In a preflow \(f\), if \(\text{excess}(v) > 0\), then there is a path from \(v\) to \(s\) in \(G_f\)

**Property 2:** At any point in the algorithm, for any \(v\), \(h(v) \leq 2n - 1\)

**Property 3:** Any node can be relabeled at most \(2n\) times in the algorithm

**Proof:** Labels never decrease, start at 0, increase by at least 1 per relabel, and can only go up to \(2n - 1\)

**Push\((v, w)\):**
- Applies if \(\text{excess}(v) > 0\), \(h(w) < h(v)\)
- \(q = \min(\text{excess}(v), c_f(v, w))\)
- Add \(q\) to \(f(v, w)\)

**Relabel\((v)\):**
- Applies if \(\text{excess}(v) > 0\) and for all \(w\) s.t. \((v, w)\) in \(E_f\), \(h(w) \geq h(v)\)
- Increase \(h(v)\) by 1
Preflow Push: #Relabels

Start with labeling: $h(s) = n, h(t) = 0, h(v) = 0$, for other $v$
Start with preflow $f$: $f(e) = c(e)$ for $e = (s, v)$, $f(e) = 0$, ow

While there is a node (other than $t$) with positive excess
   Pick a node $v$ with $\text{excess}(v) > 0$
   If there is an edge $(v, w)$ in $E_f$ s.t. $\text{push}(v, w)$ applies
      Push($v, w$)
   Else
      Relabel($v$)

Property 1: In a preflow $f$, if $\text{excess}(v) > 0$, then there is a path from $v$ to $s$ in $G_f$

Property 2: At any point in the algorithm, for any $v$, $h(v) \leq 2n - 1$

Property 3: Any node can be relabeled at most $2n$ times in the algorithm

Total #relabel operations = $O(n^2)$

Push($v, w$):
Applies if $\text{excess}(v) > 0$, $h(w) < h(v)$
   $q = \min(\text{excess}(v), c_r(v, w))$
   Add $q$ to $f(v, w)$

Relabel($v$):
Applies if $\text{excess}(v) > 0$ and for all $w$ s.t. $(v, w)$ in $E_f$, $h(w) \geq h(v)$
   Increase $h(v)$ by 1

Diagram:
- Nodes $v, w$
- Edges between nodes
- Height $h$ axis
- Nodes labeled with $h$ values
Running Time Analysis: Outline

1. How many Push Ops? Relabel Ops?

2. How to implement Push and Relabel Ops efficiently?
Running Time Analysis: Outline

1. How many Relabel Ops? How many Push Ops?

Two types of Push Ops:

**Saturating Pushes:** \((v, w)\) is saturated after \(\text{push}(v, w)\)
   
   Same edge can’t be pushed on until a relabel (we will see why!)

**Non-saturating Pushes:** \(\text{excess}(v) = 0\) after \(\text{push}(v, w)\)
Preflow Push: \#Pushes

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)

Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess
  Pick a node \( v \) with \( \text{excess}(v) > 0 \)
  If there is an edge \((v, w)\) in \( E_f \) s. t. \( \text{push}(v, w) \) applies
    Push\((v, w)\)
  Else
    Relabel\((v)\)

Two kinds of Pushes:
  **Saturating:** \((v, w)\) is not in \( G_f \) after push
  **Nonsaturating:** \( \text{excess}(v) \) becomes 0 after push

**Property 1:** There are at most \( 2mn \) saturating pushes

**Proof:** For a fixed edge \((v, w)\), after a saturating push, we can only push along \((v, w)\) again once \( v \) is relabeled

**Push\((v, w)\):**
Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
  \( q = \min(\text{excess}(v), c_f(v,w)) \)
  Add \( q \) to \( f(v, w) \)

**Relabel\((v)\):**
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s. t. \((v, w)\) in \( E_f \), \( h(w) \geq h(v) \)
  Increase \( h(v) \) by 1
Preflow Push: #Pushes

Start with labeling: $h(s) = n, h(t) = 0, h(v) = 0$, for other $v$
Start with preflow $f$: $f(e) = c(e)$ for $e = (s, v), f(e) = 0$, otherwise

While there is a node (other than $t$) with positive excess
Pick a node $v$ with $\text{excess}(v) > 0$
If there is an edge $(v, w)$ in $E_f$ s.t. push$(v, w)$ applies
  Push$(v, w)$
Else
  Relabel$(v)$

Two kinds of Pushes:
- **Saturating**: $(v, w)$ is not in $G_f$ after push
- **Nonsaturating**: $\text{excess}(v)$ becomes 0 after push

**Property 1**: There are at most $2mn$ saturating pushes

**Proof**: For a fixed edge $(v, w)$, after a saturating push, we can only push along $(v, w)$ again once $v$ is relabeled

**Push**$(v, w)$:
  Applies if $\text{excess}(v) > 0$, $h(w) < h(v)$
  $q = \min(\text{excess}(v), c_f(v, w))$
  Add $q$ to $f(v, w)$

**Relabel**$(v)$:
  Applies if $\text{excess}(v) > 0$ and for all $w$ s.t $(v, w)$ in $E_f$, $h(w) \geq h(v)$
  Increase $h(v)$ by 1

(v,w) disappears from $G_f$ after saturating push, appears only after $w$ to $v$ push
Preflow Push: \#Pushes

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess
Pick a node \( v \) with \( \text{excess}(v) > 0 \)
If there is an edge \((v, w)\) in \( E_f \) s. t. \( \text{push}(v, w) \) applies
Push\((v, w)\)
Else
Relabel\((v)\)

Two kinds of Pushes:
- **Saturating**: \((v, w)\) is not in \( G_f \) after push
- **Nonsaturating**: \( \text{excess}(v) \) becomes 0 after push

**Property 1**: There are at most \( 2mn \) saturating pushes

**Proof**: For a fixed edge \((v, w)\), after a saturating push, we can only push along \((v, w)\) again once \( v \) is relabeled
\#relabels of \( v \) \( \leq 2n \)

**Push\((v, w)\):**
- Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
- \( q = \min(\text{excess}(v), c_f(v,w)) \)
- Add \( q \) to \( f(v, w) \)

**Relabel\((v)\):**
- Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t \((v, w)\) in \( E_f \), \( h(w) \geq h(v) \)
- Increase \( h(v) \) by 1

\((v,w)\) disappears from \( G_f \) after saturating push, appears only after \( w \) to \( v \) push
Preflow Push: \#Pushes

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f : f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess
- Pick a node \( v \) with excess\((v) > 0\)
  - If there is an edge \((v, w)\) in \( E_f \) s.t. push\((v, w)\) applies
    - Push\((v, w)\)
  - Else
    - Relabel\((v)\)

Two kinds of Pushes:
  - **Saturating:** \((v, w)\) is not in \( G_f \) after push
  - **Nonsaturating:** excess\((v)\) becomes 0 after push

**Property 1:** There are at most \( 2mn \) saturating pushes

**Proof:** For a fixed edge \((v, w)\), after a saturating push, we can only push along \((v, w)\) again once \( v \) is relabeled
- \#relabels of \( v \) \( \leq 2n \)
- \#saturating pushes along \((v, w)\) \( \leq 2n \)

**Push**\((v, w)\):  
Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
- \( q = \min(\text{excess}(v), c_f(v, w)) \)
- Add \( q \) to \( f(v, w) \)

**Relabel**\((v)\):  
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t. \((v, w)\) in \( E_f \), \( h(w) \geq h(v) \)
- Increase \( h(v) \) by 1

\[ (v, w) \text{ disappears from } G_f \text{ after saturating push, appears only after } w \text{ to } v \text{ push} \]
**Preflow Push: \#Pushes**

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v), f(e) = 0, \text{ow} \)

While there is a node (other than \( t \)) with positive excess
- Pick a node \( v \) with \( \text{excess}(v) > 0 \)
- If there is an edge \((v, w)\) in \( E_f \) s. t. push\((v, w)\) applies
  - Push\((v, w)\)
- Else
  - Relabel\((v)\)

Two kinds of Pushes:
- **Saturating:** \((v,w)\) is not in \( G_f \) after push
- **Nonsaturating:** \( \text{excess}(v) \) becomes 0 after push

**Property 1:** There are at most \( 2mn \) saturating pushes

**Proof:** For a fixed edge \((v,w)\), after a saturating push, we can only push along \((v, w)\) again once \( v \) is relabeled
- \#relabels of \( v \) \( \leq 2n \)
- \#saturating pushes along \((v, w)\) \( \leq 2n \)
- \#saturating pushes along all \( m \) edges \( \leq 2nm \)

### Push\((v, w)\):
Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
- \( q = \min(\text{excess}(v), c_f(v, w)) \)
- Add \( q \) to \( f(v, w) \)

### Relabel\((v)\):
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s. t. \((v, w)\) in \( E_f \), \( h(w) \geq h(v) \)
- Increase \( h(v) \) by 1

\( (v,w) \) disappears from \( G_f \) after saturating push, appears only after \( w \) to \( v \) push
Running Time Analysis: Outline

1. How many Relabel Ops? How many Push Ops?

Two types of Push Ops:

**Saturating Pushes:** \((v, w)\) is saturated after \(\text{push}(v, w)\)
   - Same edge can’t be pushed on until a relabel

**Non-saturating Pushes:** \(\text{excess}(v) = 0\) after \(\text{push}(v, w)\)
   - Harder to bound. Need to use a potential function argument
Two kinds of Pushes:
- **Saturating:** \((v,w)\) is not in \(G_f\) after push
- **Nonsaturating:** \(\text{excess}(v)\) becomes 0 after push

Property 0: There are \(\leq 2n^2\) relabels
Property 1: There are \(\leq 2mn\) saturating pushes
Property 2: There are \(\leq 4mn^2\) non-saturating pushes

**Push\((v, w)\):**
Applies if \(\text{excess}(v) > 0\), \(h(w) < h(v)\)
\[
q = \min(\text{excess}(v), c_f(v,w))
\]
Add \(q\) to \(f(v, w)\)

**Relabel\((v)\):**
Applies if \(\text{excess}(v) > 0\) and for all \(w\) s.t \((v, w)\) in \(E_f\), \(h(w) \geq h(v)\)
Increase \(h(v)\) by 1
Preflow Push: #Pushes

Two kinds of Pushes:
- **Saturating**: \((v, w)\) is not in \(G_f\) after push
- **Nonsaturating**: \(\text{excess}(v)\) becomes 0 after push

Property 0: There are \(\leq 2n^2\) relabels
Property 1: There are \(\leq 2mn\) saturating pushes
Property 2: There are \(\leq 4mn^2\) non-saturating pushes

Proof: Define a potential function \(G(f, h)\):
\[
G(f, h) = \sum_{v: \text{excess}(v) > 0} h(v)
\]

Push\((v, w)\):
Applies if \(\text{excess}(v) > 0, h(w) < h(v)\)
\[q = \min(\text{excess}(v), c_f(v, w))\]
Add \(q\) to \(f(v, w)\)

Relabel\((v)\):
Applies if \(\text{excess}(v) > 0\) and for all \(w\) s.t \((v, w)\) in \(E_f\), \(h(w) \geq h(v)\)
Increase \(h(v)\) by 1
**Preflow Push: #Pushes**

**Two kinds of Pushes:**
- **Saturating:** \( (v,w) \) is not in \( G_f \) after push
- **Nonsaturating:** \( \text{excess}(v) \) becomes 0 after push

**Property 0:** There are \( \leq 2n^2 \) relabels
**Property 1:** There are \( \leq 2mn \) saturating pushes
**Property 2:** There are \( \leq 4mn^2 \) non-saturating pushes

**Proof:** Define a potential function \( G(f, h) \):
\[
G(f, h) = \sum_{v: \text{excess}(v) > 0} h(v)
\]
Initially, \( G(f, h) = 0 \)
At any time, \( G(f, h) \geq 0 \)

**Push\((v, w)\):**
- Applies if \( \text{excess}(v) > 0 \), \( h(w) < h(v) \)
- \( q = \min(\text{excess}(v), c_f(v,w)) \)
- Add \( q \) to \( f(v, w) \)

**Relabel\((v)\):**
- Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t \( (v, w) \) in \( E_f \), \( h(w) \geq h(v) \)
- Increase \( h(v) \) by 1
Preflow Push: #Pushes

Two kinds of Pushes:

**Saturating:** \((v,w)\) is not in \(G_f\) after push

**Nonsaturating:** \(\text{excess}(v)\) becomes 0 after push

**Property 0:** There are \(\leq 2n^2\) relabels

**Property 1:** There are \(\leq 2mn\) saturating pushes

**Property 2:** There are \(\leq 4mn^2\) non-saturating pushes

**Proof:** Define a potential function \(G(f, h)\):

\[
G(f, h) = \sum_{v: \text{excess}(v) > 0} h(v)
\]

Initially, \(G(f, h) = 0\)

At any time, \(G(f, h) \geq 0\)

At a relabel operation, \(G(f, h)\) can increase by 1

**Push\((v, w)\):**

 Applies if \(\text{excess}(v) > 0, h(w) < h(v)\)

\[
q = \min(\text{excess}(v), c_f(v, w))
\]

Add \(q\) to \(f(v, w)\)

**Relabel\((v)\):**

 Applies if \(\text{excess}(v) > 0\) and for all \(w\) s.t \((v, w)\) in \(E_f\), \(h(w) \geq h(v)\)

Increase \(h(v)\) by 1
Preflow Push: #Pushes

Two kinds of Pushes:

- **Saturating**: (v, w) is not in G_f after push
- **Nonsaturating**: excess(v) becomes 0 after push

**Property 0**: There are <= 2n^2 relabels
**Property 1**: There are <= 2mn saturating pushes
**Property 2**: There are <= 4mn^2 non-saturating pushes

**Proof**: Define a potential function G(f, h):

\[ G(f, h) = \sum_{v : excess(v) > 0} h(v) \]

Initially, G(f, h) = 0
At any time, G(f, h) >= 0
At a relabel operation, G(f, h) can increase by 1
At a saturating push operation, G(f, h) can increase if w gets >0 excess. Total increase = h(w) <= 2n - 1

---

**Push(v, w):**
Applies if excess(v) > 0, h(w) < h(v)
q = \min\{excess(v), c_f(v, w)\}
Add q to f(v, w)

**Relabel(v):**
Applies if excess(v) > 0 and for all w s.t. (v, w) in E_f, h(w) >= h(v)
Increase h(v) by 1
Preflow Push: \#Pushes

Two kinds of Pushes:
- **Saturating**: \((v, w)\) is not in \(G_f\) after push
- **Nonsaturating**: \(\text{excess}(v)\) becomes 0 after push

**Property 0:** There are \(\leq 2n^2\) relabels
**Property 1:** There are \(\leq 2mn\) saturating pushes
**Property 2:** There are \(\leq 4mn^2\) non-saturating pushes

**Proof:** Define a potential function \(G(f, h)\):
\[
G(f, h) = \sum_{v: \text{excess}(v) > 0} h(v)
\]
Initially, \(G(f, h) = 0\)

At any time, \(G(f, h) \geq 0\)

At a relabel operation, \(G(f, h)\) can increase by 1

At a saturating push operation, \(G(f, h)\) can increase if \(w\) gets \(>0\) excess. Total increase = \(h(w) \leq 2n - 1\)

At a non-saturating push operation, \(G(f, h)\) will decrease by \(h(v)\), but may increase by \(h(w)\) if \(w\) gets \(>0\) excess
But \(h(v) > h(w)\), so \(G(f, h)\) will decrease by at least 1

**Push\((v, w)\):**
Applies if \(\text{excess}(v) > 0, h(w) < h(v)\)
\[
q = \min(\text{excess}(v), c_f(v, w))
\]
Add \(q\) to \(f(v, w)\)

**Relabel\((v)\):**
Applies if \(\text{excess}(v) > 0\) and for all \(w\) s.t \((v, w)\) in \(E_f\), \(h(w) \geq h(v)\)
Increase \(h(v)\) by 1
Preflow Push: \#Pushes

\textbf{Two kinds of Pushes:}
\begin{itemize}
\item \textbf{Saturating:} \((v,w)\) is not in \(G_f\) after push
\item \textbf{Nonsaturating:} \(\text{excess}(v)\) becomes 0 after push
\end{itemize}

\textbf{Property 0:} There are \(\leq 2n^2\) relabels
\textbf{Property 1:} There are \(\leq 2mn\) saturating pushes
\textbf{Property 2:} There are \(\leq 4mn^2\) non-saturating pushes

\textbf{Proof:} Define a potential function \(G(f, h):\)
\[
G(f, h) = \sum_{v: \text{excess}(v) > 0} h(v)
\]

Initially, \(G(f, h) = 0\)
At any time, \(G(f, h) \geq 0\)
At a relabel operation, \(G(f, h)\) can increase by 1
At a saturating push operation, \(G(f, h)\) can increase if \(w\) gets \(> 0\) excess. Total increase = \(h(w) \leq 2n - 1\)
At a non-saturating push operation, \(G(f, h)\) will decrease by \(h(v)\), but may increase by \(h(w)\) if \(w\) gets \(> 0\) excess
But \(h(v) > h(w)\), so \(G(f, h)\) will decrease by at least 1

\textbf{Push}(v, w):
Applies if \(\text{excess}(v) > 0, h(w) < h(v)\)
\[
q = \min(\text{excess}(v), cf(v,w))
\]
Add \(q\) to \(f(v, w)\)

\textbf{Relabel}(v):
Applies if \(\text{excess}(v) > 0\) and for all \(w\) s.t \((v, w)\) in \(E_f\), \(h(w) \geq h(v)\)
Increase \(h(v)\) by 1

Total increase from relabels \(\leq 2n^2\)
Preflow Push: \#Pushes

**Two kinds of Pushes:**

**Saturating:** \((v,w)\) is not in \(G_f\) after push  
**Nonsaturating:** \(\text{excess}(v)\) becomes 0 after push

**Property 0:** There are \(\leq 2n^2\) relabels  
**Property 1:** There are \(\leq 2mn\) saturating pushes  
**Property 2:** There are \(\leq 4mn^2\) non-saturating pushes

**Proof:** Define a potential function \(G(f, h)\):

\[
G(f, h) = \sum_{v: \text{excess}(v) > 0} h(v)
\]

Initially, \(G(f, h) = 0\)

At any time, \(G(f, h) \geq 0\)

At a relabel operation, \(G(f, h)\) can increase by 1

At a saturating push operation, \(G(f, h)\) can increase if \(w\) gets \(>0\) excess. Total increase = \(h(w) \leq 2n - 1\)

At a non-saturating push operation, \(G(f, h)\) will decrease by \(h(v)\), but may increase by \(h(w)\) if \(w\) gets \(>0\) excess  
But \(h(v) > h(w)\), so \(G(f, h)\) will decrease by at least 1

---

**Push\((v, w)\):**
Applies if \(\text{excess}(v) > 0\), \(h(w) < h(v)\)  
\(q = \min(\text{excess}(v), c_f(v, w))\)  
Add \(q\) to \(f(v, w)\)

**Relabel\((v)\):**
Applies if \(\text{excess}(v) > 0\) and for all \(w\) s.t \((v, w)\) in \(E_f\), \(h(w) \geq h(v)\)  
Increase \(h(v)\) by 1  

Total increase from relabels \(\leq 2n^2\)
Total increase from saturating pushes \(\leq 2mn(2n - 1)\)
Preflow Push: #Pushes

Two kinds of Pushes:

Saturating: \((v, w)\) is not in \(G_f\) after push

Nonsaturating: \(\text{excess}(v)\) becomes 0 after push

Property 0: There are \(\leq 2n^2\) relabels

Property 1: There are \(\leq 2mn\) saturating pushes

Property 2: There are \(\leq 4mn^2\) non-saturating pushes

Proof: Define a potential function \(G(f, h)\):

\[
G(f, h) = \sum_{v: \text{excess}(v) > 0} h(v)
\]

Initially, \(G(f, h) = 0\)

At any time, \(G(f, h) \geq 0\)

At a relabel operation, \(G(f, h)\) will increase by 1

At a saturating push operation, \(G(f, h)\) can increase if \(w\) gets >0 excess. Total increase = \(h(w) \leq 2n - 1\)

At a non-saturating push operation, \(G(f, h)\) will decrease by \(h(v)\), but may increase by \(h(w)\) if \(w\) gets >0 excess

But \(h(v) > h(w)\), so \(G(f, h)\) will decrease by at least 1

Total increase from relabels \(\leq 2n^2\)

Total increase from saturating pushes

pushes \(\leq 2mn(2n - 1)\)

(#non-saturating pushes) \(\times 1\)

\(\leq \) Total decrease from such pushes

\(\leq \) total increase from anything else

\(\leq 2n^2 + 2mn(2n - 1) = 4mn^2\)

Push\((v, w)\):

Applies if \(\text{excess}(v) > 0\), \(h(w) < h(v)\)

\[q = \min(\text{excess}(v), c_f(v, w))\]

Add \(q\) to \(f(v, w)\)

Relabel\((v)\):

Applies if \(\text{excess}(v) > 0\) and for all \(w\) s.t \((v, w)\) in \(E_f\), \(h(w) \geq h(v)\)

Increase \(h(v)\) by 1
Preflow Push: #Pushes

Two kinds of Pushes:
- **Saturating**: \((v,w)\) is not in \(G_f\) after push
- **Nonsaturating**: \(\text{excess}(v)\) becomes 0 after push

Property 0: There are \(\leq 2n^2\) relabels

Property 1: There are \(\leq 2mn\) saturating pushes

Property 2: There are \(\leq 4mn^2\) non-saturating pushes

Proof: Define a potential function \(G(f, h)\):
\[
G(f, h) = \sum_{v: \text{excess}(v) > 0} h(v)
\]
Initially, \(G(f, h) = 0\)
At any time, \(G(f, h) \geq 0\)
At a relabel operation, \(G(f, h)\) can increase by 1
At a saturating push operation, \(G(f, h)\) can increase if \(w\) gets \(> 0\) excess. Total increase = \(h(w) \leq 2n - 1\)
At a non-saturating push operation, \(G(f, h)\) will decrease by \(h(v)\), but may increase by \(h(w)\) if \(w\) gets \(> 0\) excess. But \(h(v) > h(w)\), so \(G(f, h)\) will decrease by at least 1

\[
\text{Total increase from relabels} \leq 2n^2
\]
\[
\text{Total increase from saturating pushes} \leq 2mn(2n - 1)
\]
\[
(\#\text{non-saturating pushes}) \times 1 \leq \text{Total decrease from such pushes} \leq \text{total increase from anything else} \leq 2n^2 + 2mn(2n - 1) = 4mn^2
\]

#Non-saturating Pushes \(\leq 4mn^2\)

---

**Push\((v, w)\):**
Applies if \(\text{excess}(v) > 0\), \(h(w) < h(v)\)
\[q = \min(\text{excess}(v), c_f(v, w))\]
Add \(q\) to \(f(v, w)\)

**Relabel\((v)\):**
Applies if \(\text{excess}(v) > 0\) and for all \(w\) s.t \((v, w)\) in \(E_f\), \(h(w) \geq h(v)\)
Increase \(h(v)\) by 1
Preflow Push: #Pushes

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)

Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), otherwise

While there is a node (other than \( t \)) with positive excess
  Pick a node \( v \) with \( \text{excess}(v) > 0 \)
  If there is an edge \( (v, w) \) in \( E_f \) s.t. push\((v, w)\) applies
    Push\((v, w)\)
  Else
    Relabel\((v)\)

Two kinds of Pushes:
  **Saturating:** \((v, w)\) is not in \( G_f \) after push
  **Nonsaturating:** \( \text{excess}(v) \) becomes 0 after push

**Property 0:** There are at most \( 2n^2 \) relabels
**Property 1:** There are at most \( 2mn \) saturating pushes
**Property 2:** There are at most \( 4mn^2 \) non-saturating pushes

**Total #Pushes:** \( O(mn^2) \)
Running Time Analysis: Outline

1. How many Push Ops? Relabel Ops?

2. How to implement Push and Relabel Ops efficiently?
**Preflow Push: Data Structures**

Start with labeling:
\[ h(s) = n, h(t) = 0, h(v) = 0, \text{ for other } v \]

Start with preflow:
\[ f(e) = c(e) \text{ for } e = (s, v), f(e) = 0, \text{ otherwise} \]

While there is a node (other than \( t \)) with positive excess:
- Pick a node \( v \) with \( \text{excess}(v) > 0 \)
- If there is an edge \((v, w)\) in \( E_f \) s.t. \( \text{push}(v, w) \) applies
  \[ \text{Push}(v, w) \]
- Else
  \[ \text{Relabel}(v) \]

### Push\((v, w)\):
- Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
- \( q = \min(\text{excess}(v), c_f(v, w)) \)
- Add \( q \) to \( f(v, w) \)

### Relabel\((v)\):
- Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t. \((v, w)\) in \( E_f \), \( h(w) \geq h(v) \)
- Increase \( h(v) \) by 1

1. For each label, use a list to maintain nodes with excess > 0

<table>
<thead>
<tr>
<th>Label Lists</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h=0 )</td>
</tr>
<tr>
<td>( h=1 )</td>
</tr>
<tr>
<td>....</td>
</tr>
</tbody>
</table>

---
1. For each label, use a list to maintain nodes with excess > 0
   Time to select a v with excess(v) > 0: O(1)

Start with labeling:
\[ h(s) = n, h(t) = 0, h(v) = 0, \text{ for other } v \]

Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0, \) ow

While there is a node (other than t) with positive excess
   Pick a node \( v \) with excess(v) > 0
   If there is an edge \( (v, w) \) in \( E_f \) s.t. \( \text{push}(v, w) \) applies
       \( \text{Push}(v, w) \)
   Else
       \( \text{Relabel}(v) \)

**Push(v, w):**
Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
   \( q = \min(\text{excess}(v), c_f(v,w)) \)
   Add \( q \) to \( f(v, w) \)

**Relabel(v):**
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t \( (v, w) \) in \( E_f, \) \( h(w) >= h(v) \)
   Increase \( h(v) \) by 1
Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)

Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v), f(e) = 0, \) ow

While there is a node (other than \( t \)) with positive excess
   Pick a node \( v \) with \( \text{excess}(v) > 0 \)
   If there is an edge \( (v, w) \) in \( E_f \) s. t. push\((v, w)\) applies
      Push\((v, w)\)
   Else
      Relabel\((v)\)

### Push\((v, w)\):
Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
   \( q = \min(\text{excess}(v), c_f(v, w)) \)
   Add \( q \) to \( f(v, w) \)

### Relabel\((v)\):
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s. t \( (v, w) \) in \( E_f \), \( h(w) \geq h(v) \)
   Increase \( h(v) \) by 1

1. For each label, use a list to maintain nodes with \( \text{excess} > 0 \)
   Time to select a \( v \) with \( \text{excess}(v) > 0 \): \( O(1) \)
   Time to insert or delete: \( O(1) \)

Label Lists

<table>
<thead>
<tr>
<th>Label</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>h=0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h=1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

....
Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), otherwise

While there is a node (other than \( t \)) with positive excess
Pick a node \( v \) with \( \text{excess}(v) > 0 \)
If there is an edge \((v, w) \) in \( E_f \) s.t. \( \text{push}(v, w) \) applies
Push(\( v, w \))
Else
Relabel(\( v \))

1. For each label, use a list to maintain nodes with excess > 0
   Time to select a \( v \) with \( \text{excess}(v) > 0 \): \( O(1) \)
   Time to insert or delete: \( O(1) \)

2. For each \( v \), maintain all \((v,w)\) in \( E_f \) in an adjacency list

**Push(\( v, w \))**:
Applies if \( \text{excess}(v) > 0 \), \( h(w) < h(v) \)
\( q = \min(\text{excess}(v), c_f(v,w)) \)
Add \( q \) to \( f(v, w) \)

**Relabel(\( v \))**:
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t \((v, w) \) in \( E_f \), \( h(w) \geq h(v) \)
Increase \( h(v) \) by 1

---

**Label Lists**

\( h = 0 \)

\( h = 1 \)

... 

\( P(v) \)

\( \text{adj-list}(v) \)
### Preflow Push: Data Structures

1. For each label, use a list to maintain nodes with excess > 0
   - Time to select a v with excess(v) > 0: O(1)
   - Time to insert or delete: O(1)

2. For each v, maintain all (v, w) in E_f in an adjacency list
   - Keep a pointer P(v) to the next edge we can push on

**Push(v, w):**
- Applies if excess(v) > 0, h(w) < h(v)
- q = min(excess(v), cf(v, w))
- Add q to f(v, w)

**Relabel(v):**
- Applies if excess(v) > 0 and for all w s.t. (v, w) in E_f, h(w) >= h(v)
- Increase h(v) by 1

**Label Lists**

```
<table>
<thead>
<tr>
<th>Label</th>
<th>h=0</th>
<th>h=1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
P(v)
```

adj-list(v)
Preflow Push: Data Structures

1. For each label, use a list to maintain nodes with excess > 0
   Time to select a v with excess(v) > 0: O(1)
   Time to insert or delete: O(1)

2. For each v, maintain all (v,w) in E_f in an adjacency list
   Keep a pointer P(v) to the next edge we can push on
   If excess(v) = 0, P(v) stays on the current edge

Start with labeling:
\text{h}(s) = n, \text{h}(t) = 0, \text{h}(v) = 0, \text{for other v}

Start with preflow f: \text{f}(e) = c(e) for e = (s, v), \text{f}(e) = 0, ow

While there is a node (other than t) with positive excess
   Pick a node v with excess(v) > 0
   If there is an edge (v, w) in E_f s. t. push(v, w) applies
      Push(v, w)
   Else
      Relabel(v)

\textbf{Push}(v, w):
Applies if excess(v) > 0, h(w) < h(v)
q = \min(\text{excess}(v), c_f(v,w))
Add q to f(v, w)

\textbf{Relabel}(v):
Applies if excess(v) > 0 and for all w s.t. (v, w) in E_f, h(w) >= h(v)
Increase h(v) by 1

\textbf{Label Lists}

- h=0
- h=1
- ....
- P(v)
- adj-list(v)
Preflow Push: Data Structures

1. For each label, use a list to maintain nodes with excess > 0
   Time to select a v with excess(v) > 0: $O(1)$
   Time to insert or delete: $O(1)$

2. For each v, maintain all (v, w) in $E_f$ in an adjacency list
   Keep a pointer $P(v)$ to the next edge we can push on
   If excess(v) = 0, $P(v)$ stays on the current edge
   Move $P(v)$ by 1 when current edge is saturated
   [Recall: If we push(v, w) and saturate it, then, we cannot push(v, w) again until v is relabeled]

Start with labeling:
- $h(s) = n$, $h(t) = 0$, $h(v) = 0$, for other v

Start with preflow $f$: $f(e) = c(e)$ for $e = (s, v)$, $f(e) = 0$, ow

While there is a node (other than t) with positive excess
  Pick a node v with excess(v) > 0
  If there is an edge (v, w) in $E_f$ s. t. push(v, w) applies
    Push(v, w)
  Else
    Relabel(v)

**Push(v, w):**
Applies if excess(v) > 0, $h(w) < h(v)$
$q = \min(\text{excess}(v), c_f(v, w))$
Add q to $f(v, w)$

**Relabel(v):**
Applies if excess(v) > 0 and for all
w s.t. $(v, w)$ in $E_f$, $h(w) \geq h(v)$
Increase $h(v)$ by 1

**Label Lists**

<table>
<thead>
<tr>
<th>h=0</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>h=1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

adj-list(v)
**Preflow Push: Data Structures**

1. For each label, use a list to maintain nodes with excess > 0
   - Time to select a v with excess(v) > 0: O(1)
   - Time to insert or delete: O(1)

2. For each v, maintain all (v,w) in E_f in an adjacency list
   - Keep a pointer P(v) to the next edge we can push on
   - If excess(v) = 0, P(v) stays on the current edge
   - Move P(v) by 1 when current edge is saturated
   - [Recall: If we push(v,w) and saturate it, then, we cannot push(v,w) again until v is relabeled]
   - Update P(v) and the list when v is relabeled

Start with labeling:
- h(s) = n, h(t) = 0, h(v) = 0, for other v
- Start with preflow f: f(e) = c(e) for e = (s, v), f(e) = 0, ow

While there is a node (other than t) with positive excess
- Pick a node v with excess(v) > 0
- If there is an edge (v, w) in E_f s. t. push(v, w) applies
  - Push(v, w)
- Else
  - Relabel(v)

---

**Push(v, w):**
- Applies if excess(v) > 0, h(w) < h(v)
- \( q = \min(\text{excess}(v), c_f(v, w)) \)
- Add q to f(v, w)

**Relabel(v):**
- Applies if excess(v) > 0 and for all w s.t (v, w) in E_f, h(w) >= h(v)
- Increase h(v) by 1

---

**Label Lists**

| h=0 | | | | | | |
|-----|---|---|---|---|
| h=1 | | | | | | |
| ... | | | | | | |
| P(v) | | | | | | |

**Adj-list(v)**
Preflow Push: Data Structures

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), otherwise

While there is a node (other than \( t \)) with positive excess
Pick a node \( v \) with \( \text{excess}(v) > 0 \)
If there is an edge \((v, w)\) in \( E_f \) s.t. \( \text{push}(v, w) \) applies
   \( \text{Push}(v, w) \)
Else
   \( \text{Relabel}(v) \)

1. For each label, use a list to maintain nodes with \( \text{excess} > 0 \)
   Time to select a \( v \) with \( \text{excess}(v) > 0 \): \( O(1) \)
   Time to insert or delete: \( O(1) \)

2. For each \( v \), maintain all \((v, w)\) in \( E_f \) in an adjacency list
   Keep a pointer \( \text{P}(v) \) to the next edge we can push on
   If \( \text{excess}(v) = 0 \), \( \text{P}(v) \) stays on the current edge
   Move \( \text{P}(v) \) by 1 when current edge is saturated
   [Recall: If we push \((v, w)\) and saturate it, then, we cannot
   push \((v, w)\) again until \( v \) is relabeled]
   Update \( \text{P}(v) \) and the list when \( v \) is relabeled

\textbf{Push}(v, w):
Applies if \( \text{excess}(v) > 0 \), \( h(w) < h(v) \)
\( q = \min(\text{excess}(v), c_{f}(v,w)) \)
Add \( q \) to \( f(v, w) \)

\textbf{Relabel}(v):
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t \((v, w)\) in \( E_f \), \( h(w) >= h(v) \)
Increase \( h(v) \) by 1

Time per relabel = \( O(1) \)
Preflow Push: Data Structures

Start with labeling:
\[ h(s) = n, h(t) = 0, h(v) = 0, \text{ for other } v \]

Start with preflow \( f \):
\[ f(e) = c(e) \text{ for } e = (s, v), f(e) = 0, \text{ otherwise} \]

While there is a node (other than \( t \)) with positive excess:
1. Pick a node \( v \) with \( \text{excess}(v) > 0 \)
2. If there is an edge \((v, w)\) in \( E_f \) s.t. \( \text{push}(v, w) \) applies:
   - \( \text{Push}(v, w) \)
3. Else:
   - \( \text{Relabel}(v) \)

**Push\((v, w)\):**
- Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
- \( q = \min(\text{excess}(v), c_f(v, w)) \)
- Add \( q \) to \( f(v, w) \)

**Relabel\((v)\):**
- Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t. \((v, w)\) in \( E_f \), \( h(w) \geq h(v) \)
- Increase \( h(v) \) by 1

1. For each label, use a list to maintain nodes with \( \text{excess} > 0 \)
   - Time to select a \( v \) with \( \text{excess}(v) > 0 \): \( O(1) \)
   - Time to insert or delete: \( O(1) \)

2. For each \( v \), maintain all \((v, w)\) in \( E_f \) in an adjacency list
   - Keep a pointer \( P(v) \) to the next edge we can push on
   - If \( \text{excess}(v) = 0 \), \( P(v) \) stays on the current edge
   - Move \( P(v) \) by 1 when current edge is saturated

[Recall: If we \( \text{push}(v, w) \) and saturate it, then, we cannot \( \text{push}(v, w) \) again until \( v \) is relabeled]
- Update \( P(v) \) and the list when \( v \) is relabeled

Time per relabel = \( O(1) \)
Time per push = \( O(1) \)
Preflow Push: Data Structures

Start with labeling: \( h(s) = n, h(t) = 0, h(v) = 0 \), for other \( v \)
Start with preflow \( f: f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), ow

While there is a node (other than \( t \)) with positive excess
  Pick a node \( v \) with \( \text{excess}(v) > 0 \)
  If there is an edge \( (v, w) \) in \( E_f \) s.t. \( \text{push}(v, w) \) applies
    - \( \text{Push}(v, w) \)
  Else
    - \( \text{Relabel}(v) \)

**Push**(\( v, w \)):
Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
  \( q = \min(\text{excess}(v), c_f(v, w)) \)
  Add \( q \) to \( f(v, w) \)

**Relabel**(\( v \)):
Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t \( (v, w) \in E_f \), \( h(w) >= h(v) \)
  - Increase \( h(v) \) by 1

1. For each label, use a list to maintain nodes with \( \text{excess} > 0 \)
   - Time to select a \( v \) with \( \text{excess}(v) > 0 \): \( O(1) \)
   - Time to insert or delete: \( O(1) \)
2. For each \( v \), maintain all \( (v, w) \) in \( E_f \) in an adjacency list
   - Keep a pointer \( P(v) \) to the next edge we can push on
   - If \( \text{excess}(v) = 0 \), \( P(v) \) stays on the current edge
   - Move \( P(v) \) by 1 when current edge is saturated
   - [Recall: If we \( \text{push}(v, w) \) and saturate it, then, we cannot \( \text{push}(v, w) \) again until \( v \) is relabeled]
   - Update \( P(v) \) and the list when \( v \) is relabeled

Time per relabel = \( O(1) \)
Time per push = \( O(1) \)
Time to maintain list after relabeling \( v = O(deg(v)) \)
Preflow Push: Data Structures

Start with labeling:
- \( h(s) = n, h(t) = 0, h(v) = 0 \) for other \( v \)

Start with preflow \( f_0 \):
- \( f(e) = c(e) \) for \( e = (s, v) \), \( f(e) = 0 \), otherwise

While there is a node (other than \( t \)) with positive excess:
- Pick a node \( v \) with \( \text{excess}(v) > 0 \)
- If there is an edge \((v, w) \) in \( E_f \) s.t. \( \text{push}(v, w) \) applies
  - \( \text{Push}(v, w) \)
- Else
  - \( \text{Relabel}(v) \)

**Push**(\( v, w \)):
- Applies if \( \text{excess}(v) > 0, h(w) < h(v) \)
- \( q = \min(\text{excess}(v), c_f(v, w)) \)
- Add \( q \) to \( f(v, w) \)

**Relabel**(\( v \)):
- Applies if \( \text{excess}(v) > 0 \) and for all \( w \) s.t. \((v, w) \) in \( E_f \), \( h(w) \geq h(v) \)
- Increase \( h(v) \) by 1

1. For each label, use a list to maintain nodes with \( \text{excess}(v) > 0 \)
   - Time to select a \( v \) with \( \text{excess}(v) > 0 \): \( O(1) \)
   - Time to insert or delete: \( O(1) \)

2. For each \( v \), maintain all \((v, w) \) in \( E_f \) in an adjacency list
   - Keep a pointer \( P(v) \) to the next edge we can push on
   - If \( \text{excess}(v) = 0 \), \( P(v) \) stays on the current edge
   - Move \( P(v) \) by 1 when current edge is saturated
   - [Recall: If we push \( (v, w) \) and saturate it, then, we cannot push \( (v, w) \) again until \( v \) is relabeled]
   - Update \( P(v) \) and the list when \( v \) is relabeled

Time per relabel = \( O(1) \)
Time per push = \( O(1) \)
Time to maintain list after relabeling \( v = O(\text{deg}(v)) \)

Total running time:
= \( O(m) \times \#\text{relabels}/\text{node} + O(\#\text{pushes} + \#\text{relabels}) \)
= \( O(mn) + O(mn^2) = O(mn^2) \)
1. How many Push Ops? Relabel Ops?
   \#pushes = \(O(mn^2)\), \#relabels = \(O(n^2)\)

2. How to implement Push and Relabel Ops efficiently?
   Data structure which takes: \(O(1)\) per push, \(O(\text{deg}(v))\) to relabel \(v\) once
   Total running time = \(O(mn^2)\)
Algorithms for Max-Flow

Recall: $n = \#\text{vertices, } m = \#\text{edges in } G$

• Ford-Fulkerson: Running Time = $O(m F_{\text{max}})$

• Other efficient Ford-Fulkerson Style Algorithms:
  • Edmonds-Karp: Running Time = $O(nm^2)$
  • Capacity Scaling: Running Time = $O(m^2 \log C_{\text{max}})$

• Preflow-Push: Running Time = $O(mn^2)$