CSE 202: Design and Analysis of Algorithms

Lecture 8
Next: Network Flows
Problem: Given directed graph G=(V,E), source s, sink t, edge capacities c(e), how much oil can we ship from s to t?

An s-t flow is a function: E → R such that:
- 0 <= f(e) <= 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

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 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 $s$-$t$ Cut partitions nodes into groups = $(L, R)$ s.t. $s$ in $L$, $t$ in $R$

Capacity of a cut $(L, R) = \sum_{(u,v) \in E, u \in L, v \in R} c(u, v)$

Flow across $(L, R) = \sum_{(u,v) \in E, u \in L, v \in R} f(u,v) - \sum_{(v,u) \in E, u \in L, v \in R} f(v,u)$

Property: For any flow $f$, any $s$-$t$ cut $(L, R)$, $\text{size}(f) \leq \text{capacity}(L, R)$

Proof: For any cut $(L,R)$, Flow Across $(L,R)$ cannot exceed capacity$(L,R)$

From flow conservation constraints, $\text{size}(f) = \text{flow across}(L,R) \leq \text{capacity}(L,R)$

Max-Flow $\leq$ 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.

An $s$-$t$ Cut partitions nodes into groups $= (L, R)$ such that $s$ in $L$, $t$ in $R$.

Capacity of a cut $(L, R) = \sum_{(u,v) \in E, u \in L, v \in R} c(u, v)$

Flow across $(L, R) = \sum_{(u,v) \in E, u \in L, v \in R} f(u, v) - \sum_{(v,u) \in E, u \in L, v \in R} f(v, u)$

Property: For any flow $f$, any $s$-$t$ cut $(L, R)$, $\text{size}(f) \leq \text{capacity}(L, R)$

Proof: For any cut $(L, R)$, Flow Across $(L, R)$ cannot exceed capacity $(L, R)$.
From flow conservation constraints, $\text{size}(f) = \text{flow across}(L, R) \leq \text{capacity}(L, R)$

$\text{Max-Flow} \leq \text{Min-Cut}$

In our example: Size of $f = 3$, Capacity of Cut $(s, V - s) = 3$.
Thus, a Min Cut is a certificate of optimality for a flow.
Ford-Fulkerson algorithm

**FF Algorithm:** Start with zero flow
Repeat:
  - Find a path from s to t along which flow can be increased
  - Increase the flow along that path

Example

First choose:

Next choose:

But what if we first chose:

Then we’d have to allow:

cancels out existing flow
Ford-Fulkerson, continued

**FF Algorithm:** Start with zero flow
Repeat:
  - Find a path from s to t along which flow can be increased
  - Increase the flow along that path

In any iteration, we have some flow \( f \) and we are trying to improve it. How to do this?

1: Construct a residual graph \( G_f \) (“what’s left to take?”)

\[ G_f = (V, E_f) \text{ where } E_f \subseteq E \cup E^R \]

For any \( (u,v) \) in \( E \), \( c_f(u,v) = c(u,v) - f(u,v) \)

any \( (u,v) \) in \( E^R \), \( c_f(u,v) = f(v,u) \)

[ignore edges with zero \( c_f \): don’t put them in \( E_f \)]

2: Find a path from s to t in \( G_f \)
3: Increase flow along this path, as much as possible
Example: Round 1

Construct residual graph $G_f = (V, E_f)$

$E_f \subseteq E \cup E^R$

For any $(u,v)$ in $E$ or $E^R$,

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

Find a path from $s$ to $t$ in $G_f$

Augment $f$ along this path
Construct residual graph $G_f = (V, E_f)$

$E_f \subseteq E \cup E^R$

For any $(u,v)$ in $E$ or $E^R$,

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

Find a path from $s$ to $t$ in $G_f$

Augment $f$ along this path
Example: Round 3

Construct residual graph \( G_f = (V, E_f) \)
\[
E_f \subseteq E \cup E^R
\]
For any \((u,v)\) in \(E\) or \(E^R\),
\[
c_f(u,v) = c(u,v) - f(u,v) + f(v,u)
\]
Find a path from \(s\) to \(t\) in \(G_f\)
Augment \(f\) along this path
Example: Round 3

Construct residual graph $G_f = (V, E_f)$

$E_f \subseteq E \cup E^R$

For any $(u,v)$ in $E$ or $E^R$,

c\(_f\)(u,v) = c(u,v) - f(u,v) + f(v,u)$

Find a path from $s$ to $t$ in $G_f$

Augment $f$ along this path
Analysis: Correctness

FF algorithm gives us a valid flow. But is it the maximum possible flow?

Consider final residual graph $G_f = (V, E_f)$
Let $L =$ nodes reachable from $s$ in $G_f$ and let $R =$ rest of nodes $= V - L$
So $s \in L$ and $t \in R$

Edges from $L$ to $R$ must be at full capacity
Edges from $R$ to $L$ must be empty
Therefore, flow across cut $(L,R)$ is

$$\sum_{(u,v) \in E, u \in L, v \in R} c(u, v)$$

Thus, $\text{size}(f) = \text{capacity}(L,R)$

Recall: for any flow and any cut, $\text{size}(\text{flow}) \leq \text{capacity}(\text{cut})$

Therefore $f$ is the max flow and $(L,R)$ is the min cut!

Thus, Max Flow = Min Cut
**Analysis: efficiency**

**FF Algorithm**: Start with zero flow
Repeat:
   - Find a path from s to t along which flow can be increased
   - Increase the flow along that path

A **hillclimbing** procedure

Flow size:

- max flow
- 0

Each iteration is fast (O(|E|) time).

How many iterations are needed to reach the maximum flow?

**Example**:

![Flow network](image)

#iterations can be Max Capacity
**Analysis: efficiency**

**FF Algorithm:** Start with zero flow

Repeat:
- Find a path from s to t along which flow can be increased
- Increase the flow along that path

**A hillclimbing procedure**

Flow size:

```
0
```

max flow

Each iteration is fast ($O(|E|)$ time).

How many iterations are needed to reach the maximum flow?

**Example:**

```
s 10^6 1 10^6 10^6
```

```
a
```

```
10^6 b
```

```
t
```

#iterations can be Max Capacity (with integer capacities)
An Observation: Integrality

Integral Flows: A flow $f$ is integral if $f(e)$ is an integer for all $e$

Example:

Property: If all edge capacities are integers, then, there is a max flow $f$ which is integral.

Proof: If the edge capacities are integers, then, the FF algorithm always finds an integral flow.
The FF algorithm also always finds a max flow.

Note: All max flows are not necessarily integral flows!
How to improve the efficiency?

• Ford-Fulkerson Style Algorithms:
  • Edmonds Karp
  • Capacity Scaling

• Preflow-Push
Edmonds Karp

**FF Algorithm:** Start with zero flow
Repeat:
  - Find a path from s to t along which flow can be increased
  - Increase the flow along that path

**Bad Example for FF:**

```
  s    a    b    t
  10^6 10^6 10^6 10^6
```

**Bad Path Sequence:**
(s, a, b, t), (s, b, a, t), (s, a, b, t),...

**EK Path Selection:** Find the **shortest path** along which flow can be increased
(shortest path = shortest in terms of #edges)

It can be shown that this requires only $O(|V||E|)$ iterations (Proof not in this class)
**Running Time:** $O(|V| |E|^2)$
**Edmonds Karp**

**EK Algorithm:** Start with zero flow
Repeat:
- Find the shortest path from $s$ to $t$ along which flow can be increased
- Increase the flow along that path

**Bad Example for FF:**

**Iteration 1**

**Iteration 2**
How to improve the efficiency?

- Ford-Fulkerson Style Algorithms:
  - Edmonds Karp
  - Capacity Scaling
- Preflow-Push
Capacity Scaling

**FF Algorithm:** Start with zero flow
Repeat:
  - Find a path from s to t along which flow can be increased
  - Increase the flow along that path

**Bad Example:**

```
<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>10^6</td>
<td>10^6</td>
</tr>
<tr>
<td>b</td>
<td>10^6</td>
<td>10^6</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
```

**Bad Path Sequence:**
(s, a, b, t), (s, b, a, t), (s, a, b, t), ...

**Capacity Scaling:** Find paths of high capacity first between s and t
Capacity Scaling

\(C_{\text{max}} = \text{max capacity edge. Start with } D = C_{\text{max}}\)

Start with zero flow

While \(D \geq 1\), repeat:

\(G_f(D) = \text{D-residual graph}\)

While there is a path from \(s\) to \(t\) in \(G_f(D)\) along which flow can be increased

Increase flow along that path

Update \(G_f(D)\)

\(D = D/2\)

**D-Residual Graph:** Subgraph of residual graph with only edges with capacity \(\geq D\)

**Example:** For \(f = 0\)
Capacity Scaling: Correctness

$C_{\text{max}} = \text{max capacity edge. Start with } D = C_{\text{max}}$

Start with zero flow

While $D \geq 1$, repeat:

$G_f(D) = D$-residual graph

While there is a path from $s$ to $t$ in $G_f(D)$ along which flow can be increased

Increase flow along that path

Update $G_f(D)$

$D = D/2$

**D-Residual Graph:** Subgraph of residual graph with only edges with capacity $\geq D$

**Property:** If all edge capacities are integers, algorithm outputs a max flow

**Proof:** At $D=1$, $G_f(D) = G_f$. So on termination, $G_f(D)$ has no more paths from $s$ to $t$
Capacity Scaling: Running Time

$C_{\text{max}} = \text{max capacity edge. Start with } D = C_{\text{max}}$

Start with zero flow

While $D \geq 1$, repeat:

1. $G_f(D) = \text{D-residual graph}$
2. While there is a path from $s$ to $t$ in $G_f(D)$ along which flow can be increased
   - Increase flow along that path
   - Update $G_f(D)$
3. $D = D/2$

**D-Residual Graph:** Subgraph of residual graph with only edges with capacity $\geq D$

**Property 1:** While loop 1 is executed $1 + \log_2 C_{\text{max}}$ times

**Property 2:** At the end of a $D$-scaling phase, $\text{size(max flow)} \leq \text{size(current flow)} + D|E|$

**Proof:** Let $L = \text{nodes reachable from s in } G_f(D)$ and let $R = \text{rest of nodes } = V - L$

- #edges in $G_f(D)$ in the $(L, R)$ cut = 0
- #edges in $G_f$ in the $(L,R)$ cut $\leq |E|$  
- Capacity of each such edge $< D$

Thus, $\text{size(max flow)} \leq \text{capacity}(L,R) \leq \text{size(f)} + D|E|$
Capacity Scaling: Running Time

\[ C_{\text{max}} = \text{max capacity edge}. \text{ Start with } D = C_{\text{max}} \]

Start with zero flow

While \( D \geq 1 \), repeat:

1. \( G_f(D) = \text{D-residual graph} \)
2. While there is a path from \( s \) to \( t \) in \( G_f(D) \) along which flow can be increased:
   - Increase flow along that path
   - Update \( G_f(D) \)

\[ D = D/2 \]

**D-Residual Graph**: Subgraph of residual graph with only edges with capacity \( \geq D \)

**Property 1**: While loop 1 is executed \( 1 + \log_2 C_{\text{max}} \) times

**Property 2**: At the end of a \( D \)-scaling phase, \( \text{size}(\text{max flow}) \leq \text{size}(\text{current flow}) + D|E| \)

**Property 3**: For any \( D \), \#iterations of loop 2 in the \( D \)-scaling phase \( \leq 2|E| \)

**Total Running Time**: \( O(|E|^2(1 + \log_2 C_{\text{max}})) \)

(Recall: Time to find a flow path in a residual graph = \( O(|E|) \) )
How to improve the efficiency?

- Ford-Fulkerson Style Algorithms:
  - Edmonds Karp
  - Capacity Scaling

- Preflow-Push
Preflow-Push

Main Idea:
- Each node has a label, which is a potential
- Route flow from high to low potential

Idea: Route flow along blue edges
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:

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

**Example**

\[\begin{array}{c}
\text{G} \\
\end{array}\]

\[\begin{array}{c}
\text{excess} = 1 \\
f \end{array}\]
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 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 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 push\((v, w)\) can be applied
    Push\((v, w)\)
  Else
    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: An Example

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

---

**Diagram:**

\( G \)

- \( s \) to \( a \) with 6
- \( b \) to \( a \) with 5
- \( s \) to \( b \) with 6
- \( b \) to \( t \) with 6
- \( a \) to \( t \) with 1

\( f, h \)

- \( s \) to \( a \) with 6
- \( a \) to \( t \) with 0
- \( b \) to \( a \) with 6
- \( a \) to \( t \) with 0

\( G_f \)

- \( s \) to \( a \) with 6
- \( a \) to \( t \) with 5
- \( b \) to \( a \) with 5

**Labels**

- \( s \) has label 0
- \( t \) has label 6
- \( a \) has label 6
- \( b \) has label 0

**Excesses**

- \( s \) has excess 6
- \( t \) has excess 0
- \( a \) has excess 6
- \( b \) has excess 0
Pre-Flow Push: An Example

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

\( G_f \) (before)

\( f, h \)

\( G_f \)
Pre-Flow Push: An Example

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 \( E_f \), \( h(w) \geq h(v) \)
  Increase \( h(v) \) by \( 1 \)

\( G_f \) (before)

\[
\begin{array}{cccc}
\text{s} & \text{a} & \text{b} & \text{t} \\
6 & 5 & 1 & 6 \\
6 & 5 & 1 & 6 \\
\end{array}
\]

\( f, h \)

\[
\begin{array}{cccc}
\text{s} & \text{a} & \text{b} & \text{t} \\
0 & 4 & 6 & 0 \\
6 & 6 & 6 & 6 \\
\end{array}
\]

\( G_f \)

\[
\begin{array}{cccc}
\text{s} & \text{a} & \text{b} & \text{t} \\
6 & 5 & 1 & 6 \\
6 & 5 & 1 & 6 \\
\end{array}
\]
Pre-Flow Push: An Example

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

\( G_f \) (before)

\( f, h \)

\( G_f \)
Pre-Flow Push: An Example

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
  $\text{Relabel}(v)$

**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
Pre-Flow Push: An Example

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
  - $\text{Relabel}(v)$

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

$\text{Relabel}(v)$:
- 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

$G_f$ (before)

\[
\begin{array}{c}
\text{s} & \text{a} & \text{b} & \text{t} \\
6 & 1 & 5 & 6
\end{array}
\]

\[
\begin{array}{c}
\text{s} & \text{a} & \text{b} & \text{t} \\
6 & 1 & 5 & 6
\end{array}
\]

\[
\begin{array}{c}
\text{s} & \text{a} & \text{b} & \text{t} \\
6 & 1 & 5 & 6
\end{array}
\]

$G_f$

\[
\begin{array}{c}
\text{s} & \text{a} & \text{b} & \text{t} \\
6 & 1 & 5 & 6
\end{array}
\]

$G_f$

\[
\begin{array}{c}
\text{s} & \text{a} & \text{b} & \text{t} \\
6 & 1 & 5 & 6
\end{array}
\]

$G_f$

\[
\begin{array}{c}
\text{s} & \text{a} & \text{b} & \text{t} \\
6 & 1 & 5 & 6
\end{array}
\]
Pre-Flow Push: An Example

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

\[ G_f \text{ (before)} \]

\[ f, h \]

\[ G_f \]
Pre-Flow Push: An Example

Start with labeling: \( h(s) = 0, h(t) = 0, h(v) = 0 \), 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 \( 1 \)

\( G_f \) (before)
\[
\begin{array}{c}
\text{s} & \text{a} & \text{b} & \text{t} \\
6 & 6 & 5 & 5 \\
\end{array}
\]

\( f, h \)
\[
\begin{array}{c}
\text{s} & \text{a} & \text{b} & \text{t} \\
0 & 6 & 6 & 6 \\
4 & 1 & 5 & 5 \\
\end{array}
\]

\( G_f \)
\[
\begin{array}{c}
\text{s} & \text{a} & \text{b} & \text{t} \\
6 & 6 & 5 & 5 \\
\end{array}
\]

**Labels**

**Excesses**
Pre-Flow Push: An Example

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. \( \text{push}(v, w) \) applies

\( \text{Push}(v, w) \)

Else

\( \text{Relabel}(v) \)

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

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

\( G_f \) (before)

\( f, h \)

\( G_f \)
Pre-Flow Push: An Example

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

\[ \text{Relabel}(v) \]

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

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 \( f(v, w) \)

Labels

Excesses
Pre-Flow Push: An Example

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
  \[ \text{Relabel}(v) \]

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

\[ \text{Relabel}(v): \]
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

---

\( G_f \) (before)

\( G_f \)

Labels

Excesses

\[ f, h \]
Pre-Flow Push: An Example

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

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

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
   \[ \text{Relabel}(v) \]

\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 \( E_f \), \( h(w) \geq h(v) \)
Increase \( h(v) \) by 1

\begin{itemize}
  \item \( G_f \) (before)
  \item \( f, h \)
  \item \( G_f \)
\end{itemize}
Pre-Flow Push: An Example

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

---

\( G_f \) (before)

\( s \)  
\( | \)  
\( b \)  
\( | \)  
\( t \)

\( f, h \)

\( s \)  
\( | \)  
\( b \)  
\( | \)  
\( t \)

\( G_f \)

\( s \)  
\( | \)  
\( b \)  
\( | \)  
\( t \)
Pre-Flow Push: An Example

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 \( E_f \), \( h(w) \geq h(v) \)
Increase \( h(v) \) by 1

**G_f (before)**

**f, h**

**G_f**
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
  \( \text{Relabel}(v) \)

**Pre-Flow Push: An Example**

\( \text{Push}(v, w): \)
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
Pre-Flow Push: An Example

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
  - \( \text{Relabel}(v) \)

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

\[ \text{Relabel}(v): \]
- 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

\( G_f \) (before)

\[ f, h \]

\[ G_f \]
Pre-Flow Push: An Example

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($v$)

$G_f$ (before)

---

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

---

Labels
Excesses
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**

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

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 $G_f$, $h(v) \leq h(w) + 1$
PreFlow Push: Correctness

**Invariant:** Preflow $f$ and labeling $h$ are always compatible over the Preflow-Push algorithm.

**Proof:** By induction. Initially, compatible, as $G_f$ has no $(s, v)$ edges.

Suppose $f$ and $h$ are compatible at time $t$. At time $t+1$:

- **Relabel:** Labels increase only if no downward edges in $G_f$
- **Push:** Edges in $G_f$ may be reversed. If so, as we push from high to low, a downwards edge will become an upwards edge.

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:

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)

Preflow $f$ and labeling $l$ are **compatible** if:

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

<table>
<thead>
<tr>
<th>$f$</th>
<th>$w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>s</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>v</td>
</tr>
<tr>
<td>1</td>
<td>Invalid</td>
</tr>
<tr>
<td>0</td>
<td>w t</td>
</tr>
</tbody>
</table>
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
Properties of Compatible PreFlows

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) <= h(w) + 1

Property 1: If preflow f and labeling h are compatible, then there is no s-t path in G_f

Proof: Suppose there is an s-t path in G_f

Due to compatibility,
- h(v_1) >= h(s) - 1 = n - 1
- h(v_2) >= h(v_1) - 1 >= n - 2
- ...
- h(t) = h(v_k) - 1 >= n - k > 0 (as k < n)

Contradiction!
Properties of Compatible PreFlows

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 $G_f$, $h(v) \leq h(w) + 1$

Property 1: If preflow $f$ and labeling $h$ are compatible, then there is no s-t path in $G_f$

Property 2: If flow $f$ and labeling $h$ are compatible, then $f$ is a max flow

Proof: From Property 1 and properties of max flow
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
PreFlow Push: Correctness

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
      \( \text{Relabel}(v) \)

**Invariant:** Preflow \( f \) and labeling \( h \) are always compatible over the Preflow-Push algorithm

**Fact:** When Preflow-push stops, \( f \) is a flow

**Proof:** Why does Preflow-push stop?
   - No valid push or relabel operation:
      We can always relabel or push if \( \text{excess}(v) > 0 \) for some \( v \)
   - No node \( v \) with \( \text{excess}(v) > 0 \):
      Then \( f \) is a flow!

**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

**Property 1:** If preflow \( f \) and labeling \( h \) are compatible, then there is no s-t path in \( G_f \)
**Property 2:** If flow \( f \) and labeling \( h \) are compatible, then \( f \) is a max flow
PreFlow Push: Correctness

Fact: When Preflow-push stops, $f$ is a flow

From Property 2 of compatible flows, and Invariant, $f$ is a max flow

Thus, Preflow-Push correctly outputs a maxflow

**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

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

**Fact:** When Preflow-push stops, $f$ is a flow

From Property 2 of compatible flows, and Invariant, $f$ is a max flow

Thus, Preflow-Push correctly outputs a maxflow

**Property 1:** If preflow $f$ and labeling $h$ are compatible, then there is no s-t path in $G_f$

**Property 2:** If flow $f$ and labeling $h$ are compatible, then $f$ is a max flow

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($v$)

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

• Algorithm
• Correctness
• Running Time Analysis
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 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: 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(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

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 excess(v) is never <0, excess(v)=0 for 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

A = all nodes v s.t. s is reachable from v in G_f
B = remaining nodes
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, 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:

\[
\begin{align*}
\text{h}(s) &= n, \\
\text{h}(v_{k-1}) &\leq n + 1, \\
\text{h}(v_1) &\leq n + k - 1 \\ &\leq 2n - 1
\end{align*}
\]

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 \( E_f \), \( h(w) \geq h(v) \)

If \( \text{excess}(v) = 0 \), then \( h(v) \) has not changed since the last time \( v \) had \( \text{excess} > 0 \)

Thus, \( h(v) \leq 2n - 1 \) also
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

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

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

\[ \text{Relabel}(v): \]
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

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 $\text{excess}(v) > 0$, $h(w) < h(v)$
$q = \min(\text{excess}(v), c_f(v, w))$
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

---

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($v$)
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 \(e = (s, v)\), \(f(e) = 0\) 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
  - \(\text{Relabel}(v)\)

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
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 \(\leq 4mn^2\)
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

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

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)

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

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: 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:
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
   - \( \text{Relabel}(v) \)

\( \text{Relabel}(v) \):
- 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

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

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 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

\[ \text{Total running time} = O(m) \times \#\text{relabels}/\text{node} + \]
\[ O(\#\text{pushes} + \#\text{relabels}) \]
\[ = O(mn) + O(mn^2) = O(mn^2) \]

Time per relabel = \( O(1) \)
Time per push = \( O(1) \)
Time to maintain list after relabeling \( v = O(\deg(v)) \)
Running Time Analysis: Outline

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(deg(v)) to relabel v once
   Total running time = O(mn^2)