CMPSC 465: LECTURE XIX

Maximum Flow Problem

Ke Chen

October 13, 2025

Flow network

Recall:

Definition A Flow Network is a directed graph $G = (V, E)$ s.t.:

1. Each edge $e \in E$ has a nonnegative capacity $c(e)$.
2. There is a unique source node $s \in V$.
3. There is a unique sink node $t \in V$.

Definition An $s - t$ flow in a flow network is a function f that maps each edge to a nonnegative real number ($f : E \to \mathbb{R}_{\geq 0}$) satisfying:

1. [Capacity Condition] $0 \leq f(e) \leq c(e) \quad \forall e \in E$
2. [Flow Conservation] For all $v \in V$, $v \notin \{s, t\}$:

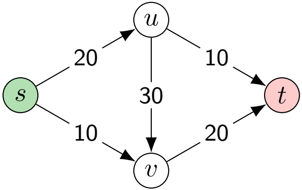
$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e).$$

The maximum flow problem

The value of a flow f is $v(f) = \sum_{e \text{ out of } s} f(e)$.

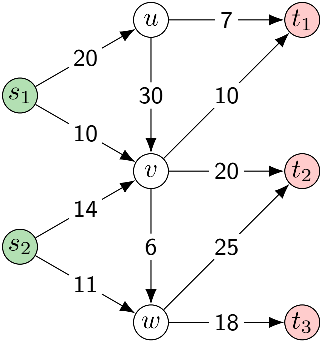
Note that $v(f) = \sum_{e \text{ into } t} f(e)$ (why?).

Max Flow Problem Given a flow network, find the flow of maximum value.



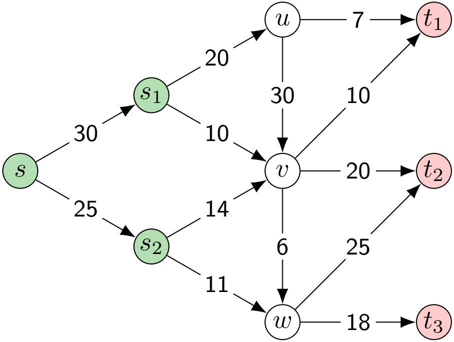
Simplifying assumptions

1. Single source, single sink.



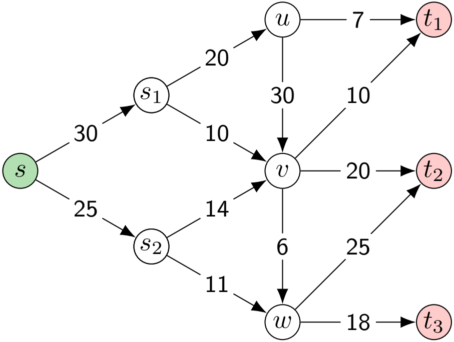
Simplifying assumptions

1. Single source, single sink.



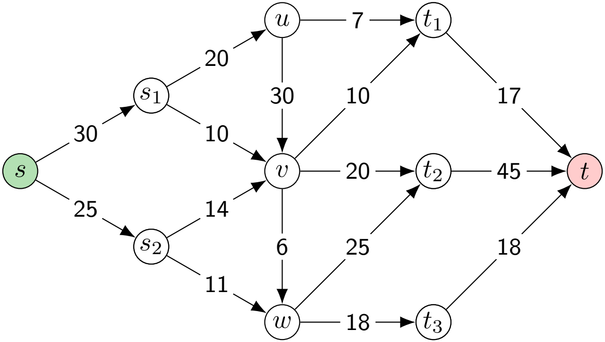
Simplifying assumptions

1. Single source, single sink.



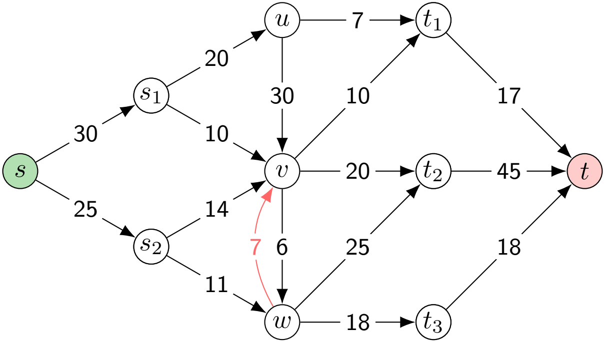
Simplifying assumptions

1. Single source, single sink.



Simplifying assumptions

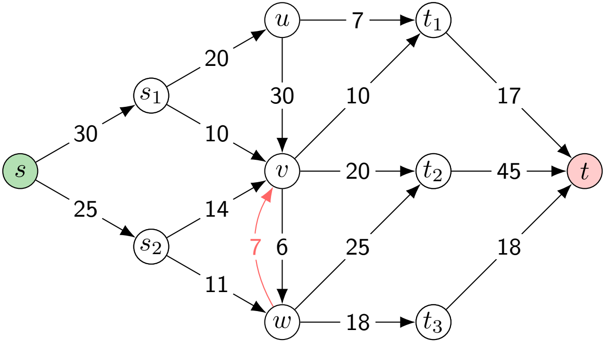
1. Single source, single sink.
2. No antiparallel edges.



Simplifying assumptions

1. Single source, single sink.
2. No antiparallel edges.

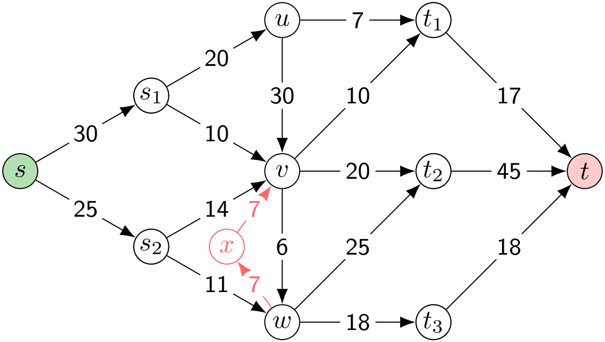
   Claim There exists a maximum flow that uses at most one edge from each pair of antiparallel edges. (Exercise)



Simplifying assumptions

1. Single source, single sink.
2. No antiparallel edges.

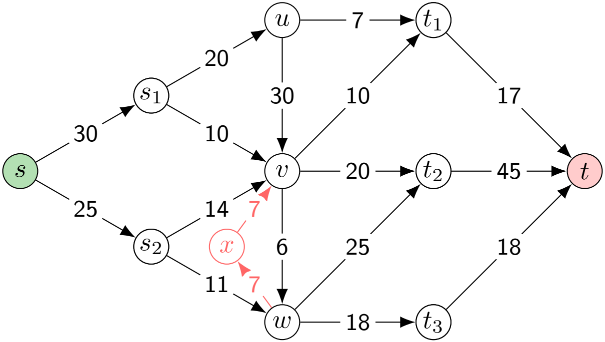
   Claim There exists a maximum flow that uses at most one edge from each pair of antiparallel edges. (Exercise)



Simplifying assumptions

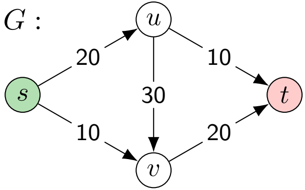
1. Single source, single sink.
2. No antiparallel edges.

   Claim There exists a maximum flow that uses at most one edge from each pair of antiparallel edges. (Exercise)
3. Integer flow capacity.



Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.

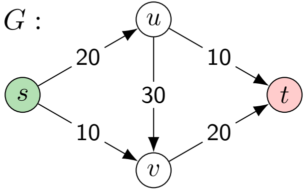


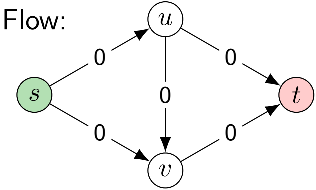
Idea

- Start with zero flow in all edges.

Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.



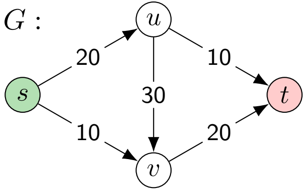


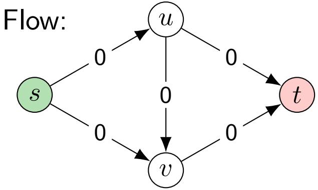
Idea

- Start with zero flow in all edges.

Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.



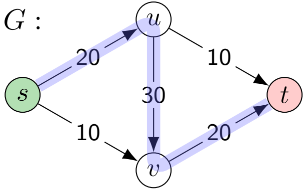


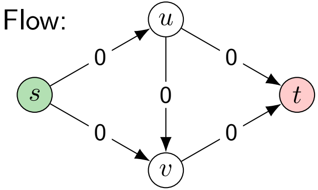
Idea

- Start with zero flow in all edges.
- Find a path from s to t and send as much flow as possible along this path.

Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.



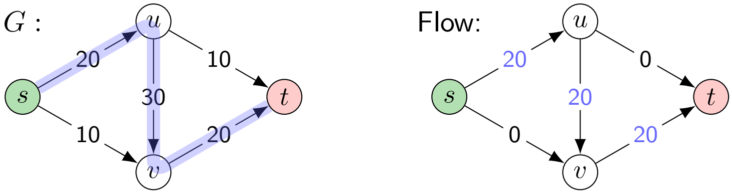


Idea

- Start with zero flow in all edges.
- Find a path from s to t and send as much flow as possible along this path.

Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.

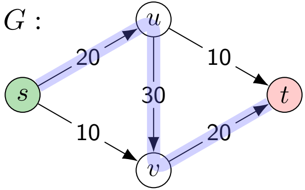


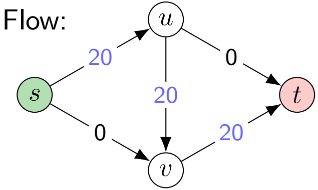
Idea

- Start with zero flow in all edges.
- Find a path from s to t and send as much flow as possible along this path.

Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.



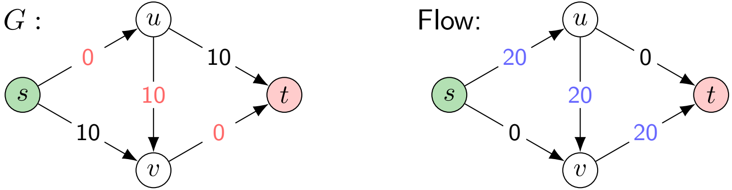


Idea

- Start with zero flow in all edges.
- Find a path from s to t and send as much flow as possible along this path.
- Adjust leftover capacities.

Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.

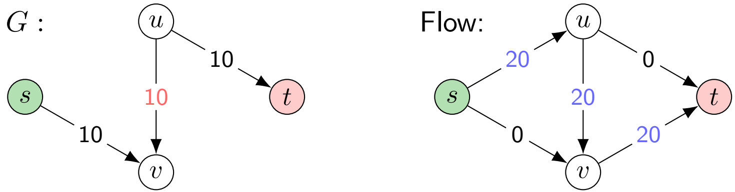


Idea

- Start with zero flow in all edges.
- Find a path from s to t and send as much flow as possible along this path.
- Adjust leftover capacities.

Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.

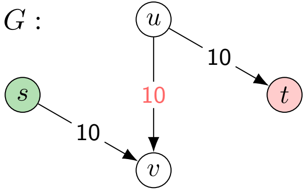


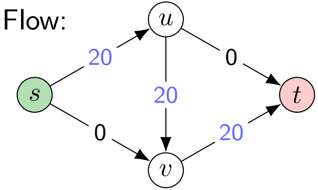
Idea

- Start with zero flow in all edges.
- Find a path from s to t and send as much flow as possible along this path.
- Adjust leftover capacities.

Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.



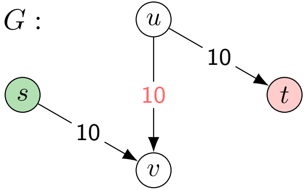


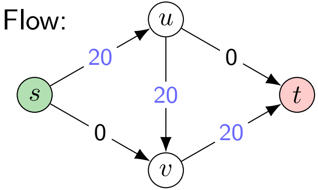
Idea

- Start with zero flow in all edges.
- Find a path from s to t and send as much flow as possible along this path.
- Adjust leftover capacities.
- Repeat while possible.

Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.





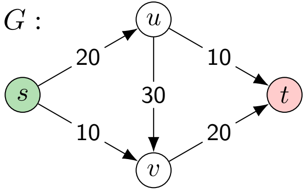
Is this optimal?

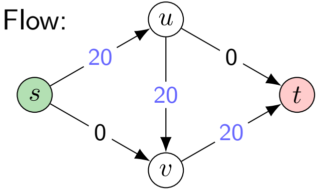
Idea

- Start with zero flow in all edges.
- Find a path from s to t and send as much flow as possible along this path.
- Adjust leftover capacities.
- Repeat while possible.

Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.





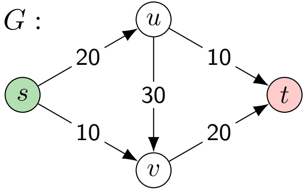
Is this optimal?

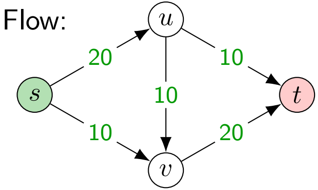
Idea

- Start with zero flow in all edges.
- Find a path from s to t and send as much flow as possible along this path.
- Adjust leftover capacities.
- Repeat while possible.

Idea for the max flow problem

Max Flow Problem Given a flow network, find the flow of maximum value.





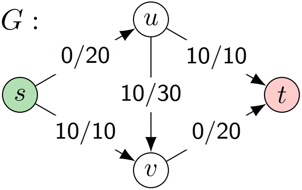
Is this optimal?

Idea

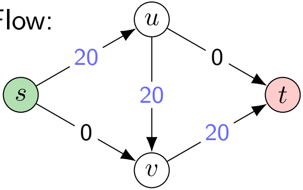
- Start with zero flow in all edges.
- Find a path from s to t and send as much flow as possible along this path.
- Adjust leftover capacities.
- Repeat while possible.

Idea for the max flow problem

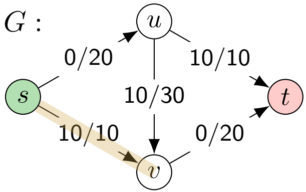
G :

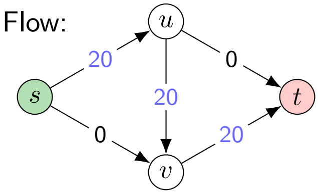


Flow:



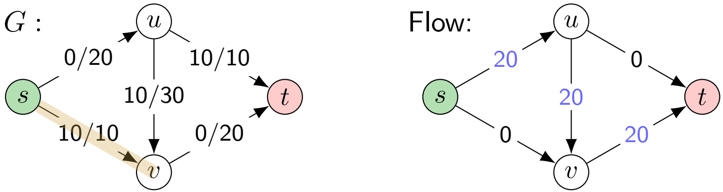
Idea for the max flow problem





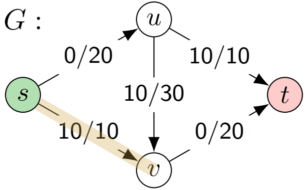
- We try to push more flow from s to t but is stuck at v.

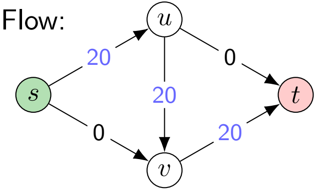
Idea for the max flow problem



- We try to push more flow from s to t but is stuck at v.
  Why?

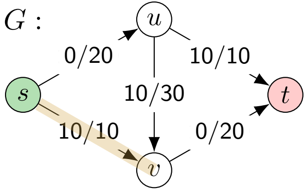
Idea for the max flow problem

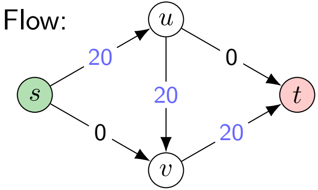




- We try to push more flow from s to t but is stuck at v. Why? 20 units of flow being routed along $u \to v \to t$.

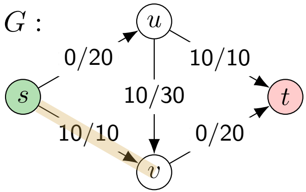
Idea for the max flow problem

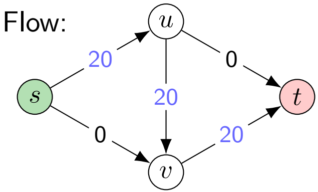




- We try to push more flow from s to t but is stuck at v. Why? 20 units of flow being routed along $u \to v \to t$.
- So we can ask u: can you re-route some flow through some other path?

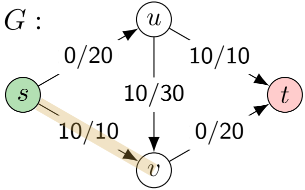
Idea for the max flow problem

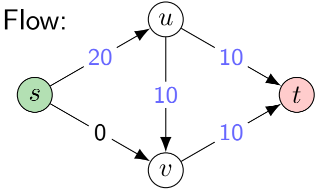




- We try to push more flow from s to t but is stuck at v.
  Why? 20 units of flow being routed along $u \to v \to t$.
- So we can ask u: can you re-route some flow through some other path?
- u answers, Yes! I can send 10 units of flow with edge (u, t).

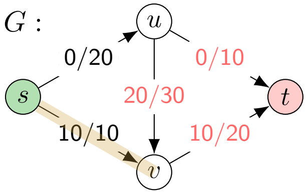
Idea for the max flow problem

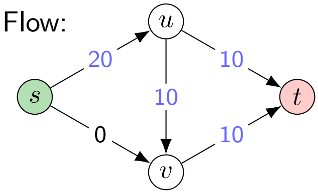




- We try to push more flow from s to t but is stuck at v.
  Why? 20 units of flow being routed along $u \to v \to t$.
- So we can ask u: can you re-route some flow through some other path?
- u answers, Yes! I can send 10 units of flow with edge (u, t).

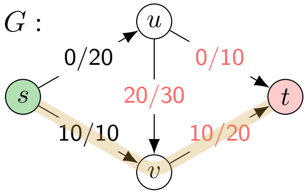
Idea for the max flow problem

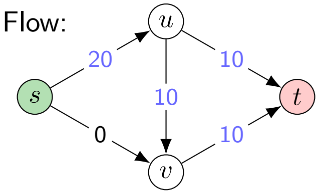




- We try to push more flow from s to t but is stuck at v. Why? 20 units of flow being routed along $u \to v \to t$.
- So we can ask u: can you re-route some flow through some other path?
- u answers, Yes! I can send 10 units of flow with edge (u, t).

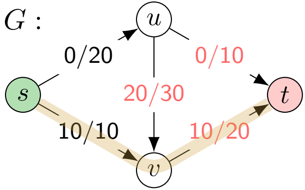
Idea for the max flow problem

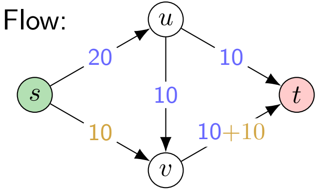




- We try to push more flow from s to t but is stuck at v. Why? 20 units of flow being routed along $u \to v \to t$.
- So we can ask u: can you re-route some flow through some other path?
- u answers, Yes! I can send 10 units of flow with edge (u, t).

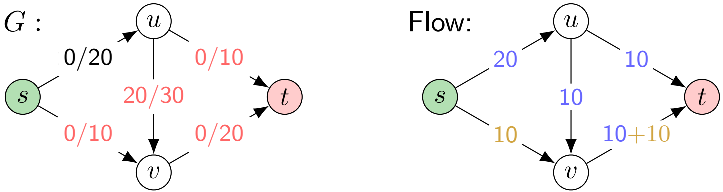
Idea for the max flow problem





- We try to push more flow from s to t but is stuck at v. Why? 20 units of flow being routed along $u \to v \to t$.
- So we can ask u: can you re-route some flow through some other path?
- u answers, Yes! I can send 10 units of flow with edge (u, t).

Idea for the max flow problem



- We try to push more flow from s to t but is stuck at v. Why? 20 units of flow being routed along $u \to v \to t$.
- So we can ask u: can you re-route some flow through some other path?
- u answers, Yes! I can send 10 units of flow with edge (u, t).

Residual graph

Idea When finding a path from s to t, it can be helpful to allow some flow being sent back.

Residual graph

Idea When finding a path from s to t, it can be helpful to allow some flow being sent back.

This motivates:

Definition Given a graph $G = (V, E)$ and a flow f, we define the residual graph G_f as follows:

- G_f has the same vertex set V as G.

Residual graph

Idea When finding a path from s to t, it can be helpful to allow some flow being sent back.

This motivates:

Definition Given a graph $G = (V, E)$ and a flow f, we define the residual graph G_f as follows:

- G_f has the same vertex set V as G.
- For an edge $e \in E$ with $f(e) < c(e)$, we include the leftover capacity $c(e) - f(e)$ in G_f [forward edge].

Residual graph

Idea When finding a path from s to t, it can be helpful to allow some flow being sent back.

This motivates:

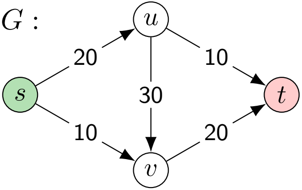
Definition Given a graph $G = (V, E)$ and a flow f, we define the residual graph G_f as follows:

- G_f has the same vertex set V as G.
- For an edge $e \in E$ with $f(e) < c(e)$, we include the leftover capacity $c(e) - f(e)$ in G_f [forward edge].
- For an edge $e = (u, v) \in E$ with $f(e) > 0$, we want to be able to "send back" the flow, so we include the reverse edge (v, u) in G_f with capacity $f(e)$ [backward edge].

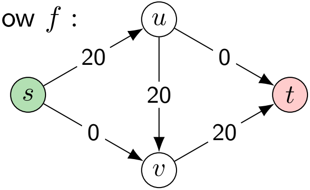
Residual graph

Example

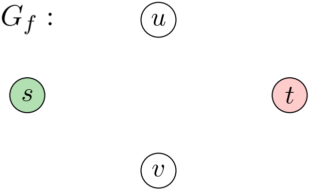
G :



Flow f :



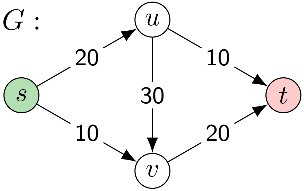
G_f :



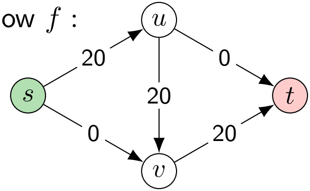
Residual graph

Example

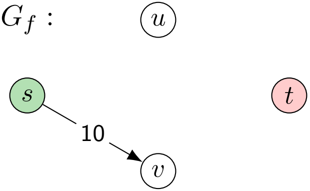
G :



Flow f :



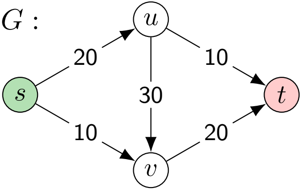
G_f :



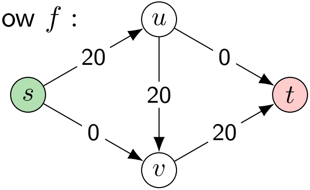
Residual graph

Example

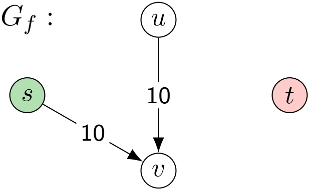
G :



Flow f :



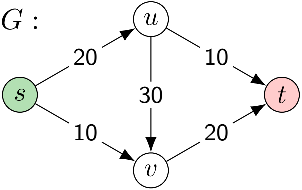
G_f :



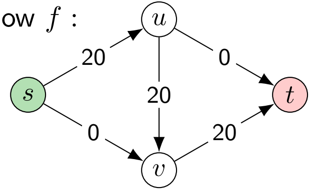
Residual graph

Example

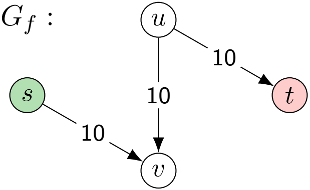
G :



Flow f :

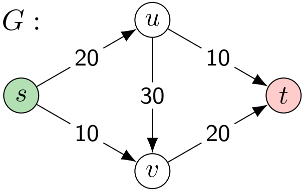


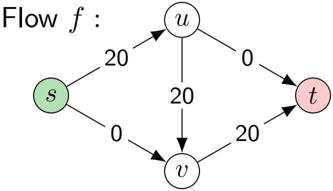
G_f :

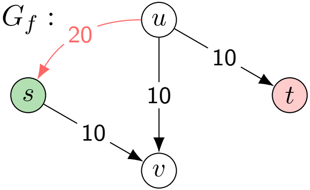


Residual graph

Example



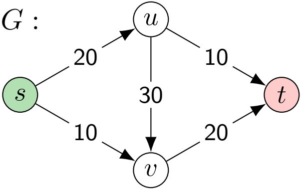




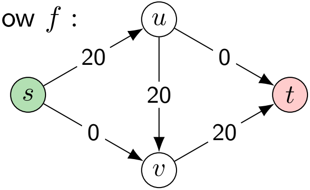
Residual graph

Example

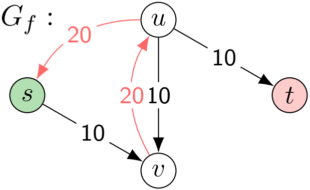
G :



Flow f :

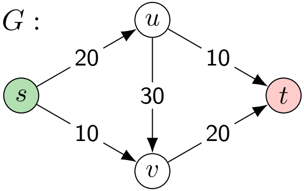


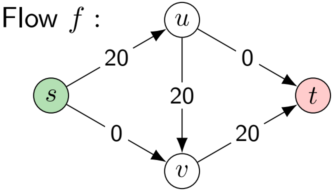
G_f :

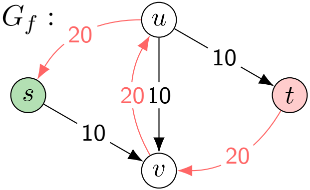


Residual graph

Example

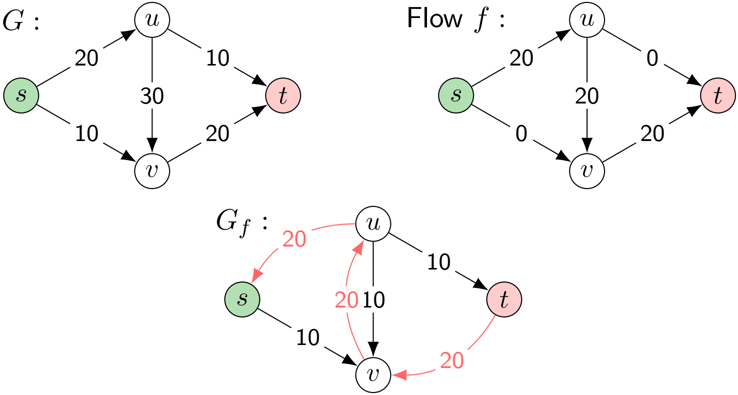






Residual graph

Example



Key Insight An $s - t$ path in G_f can be used to improve the value of the flow.

Augmenting paths on residual graph

- Let P be an $s - t$ path in G_f.

Augmenting paths on residual graph

- Let P be an $s - t$ path in G_f.
- Let $b = bottleneck(P, f)$ be the minimum residual capacity of any edge on P.

Augmenting paths on residual graph

- Let P be an $s-t$ path in G_f.
- Let $b = bottleneck(P, f)$ be the minimum residual capacity of any edge on P.
- We can improve the value of f by b:

Augmenting paths on residual graph

- Let P be an $s - t$ path in G_f.
- Let $b = bottleneck(P, f)$ be the minimum residual capacity of any edge on P.
- We can improve the value of f by b:
  - increase the flow in every forward edge by b,

Augmenting paths on residual graph

- Let P be an $s - t$ path in G_f.
- Let $b = bottleneck(P, f)$ be the minimum residual capacity of any edge on P.
- We can improve the value of f by b:
  - increase the flow in every forward edge by b,
  - decrease the flow in every backward edge by b.

Augmenting paths on residual graph

- Let P be an $s-t$ path in G_f.
- Let $b = bottleneck(P, f)$ be the minimum residual capacity of any edge on P.
- We can improve the value of f by b:
  - increase the flow in every forward edge by b,
  - decrease the flow in every backward edge by b.

Augment(P, f)

$b = bottleneck(P, f)$
foreach *edge* $(x, y) \in P$ **do**
  if (x, y) *is a forward edge* **then**
    $f(x, y) = f(x, y) + b$
  if (x, y) *is a backward edge* **then**
    $f(x, y) = f(x, y) - b$
return *f*

The Ford-Fulkerson algorithm

Input: Flow network $G = (V, E, c)$
Output: Maximum flow f
Ford-Fulkerson(G)

- Initialize $f(e) = 0$ for all $e \in E$
- $G_f = G$
- **while** *there is an* $s - t$ *path* P *in* G_f **do**
  - $f =$ Augment(P, f)
  - Build new residual graph G_f
- Output f

The Ford-Fulkerson algorithm

Input: Flow network $G = (V, E, c)$
Output: Maximum flow f
Ford-Fulkerson(G)

- Initialize $f(e) = 0$ for all $e \in E$
- $G_f = G$
- **while** *there is an* $s - t$ *path* P *in* G_f **do**
  - $f = $ Augment(P, f)
  - Build new residual graph G_f
- Output f

Correctness?

The Ford-Fulkerson algorithm

Input: Flow network $G = (V, E, c)$
Output: Maximum flow f
Ford-Fulkerson(G)

- Initialize $f(e) = 0$ for all $e \in E$
- $G_f = G$
- **while** *there is an* $s - t$ *path* P *in* G_f **do**
  - $f =$ Augment(P, f)
  - Build new residual graph G_f
- Output f

Correctness?

Time complexity?