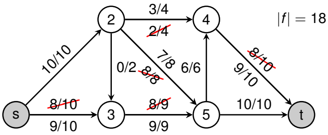
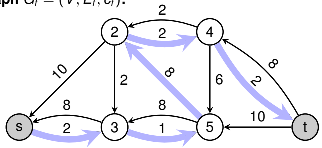
Graph $G = (V, E, c)$:



Residual Graph $G_f = (V, E_f, c_f)$:



6.6: Maximum flow

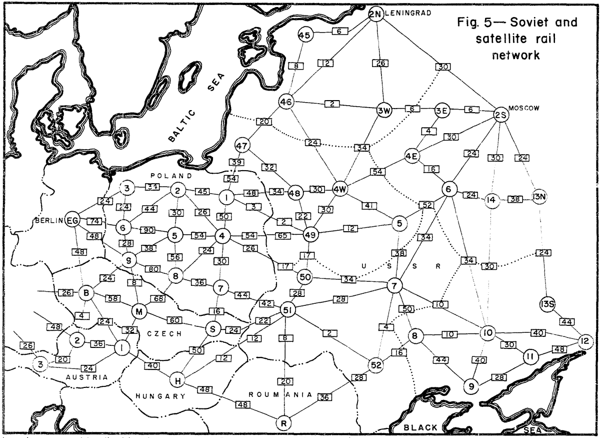
Frank Stajano
Thomas Sauerwald

Lent 2015

UNIVERSITY OF CAMBRIDGE

Outline

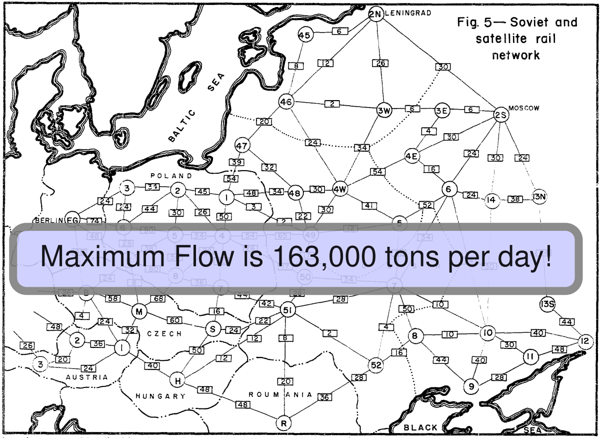
History of the Maximum Flow Problem [Harris, Ross (1955)]



Legend: —·—·— International boundary ··········· Regional boundaries of the USSR (they are included as a matter of general information)

(7) Operating divisions. Those located in Russia are believed to be accurately located. Some Russian divisions (2, 3, 4 and 13) are located in two regions and are so indicated. Divisions shown in the satellites are indicated according to the authors' best judgment, since intelligence reports are unavailable. Train capacities in Russia are for 1000-net-ton trains or their equivalent. Train capacities in Poland are for 666 net tons (or the equivalent). Train capacities in all other satellites are for 400 net tons (or the equivalent) except in East Germany. In East Germany, train capacities are those of entering lines. The numbers shown in boxes are total interdivisional capacities.

History of the Maximum Flow Problem [Harris, Ross (1955)]



Maximum Flow is 163,000 tons per day!

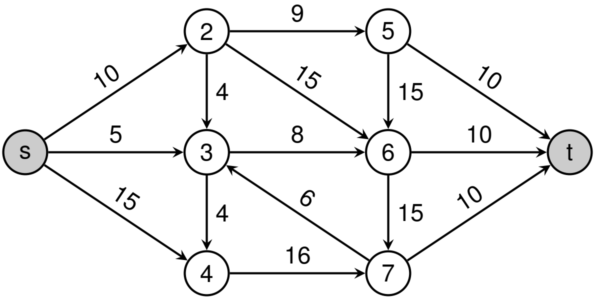
Legend: —··— International boundary ··········· Regional boundaries of the USSR (they are included as a matter of general information)

⑦ Operating divisions. Those located in Russia are believed to be accurately located. Some Russian divisions (2, 3, 4 and 13) are located in two regions and are so indicated. Divisions shown in the satellites are indicated according to the authors' best judgment, since intelligence reports are unavailable. Train capacities in Russia are for 1000-net-ton trains or their equivalent. Train capacities in Poland are for 666 net tons (or the equivalent). Train capacities in all other satellites are for 400 net tons (or the equivalent) except in East Germany. In East Germany, train capacities are those of entering lines. The numbers shown in boxes are total interdivisional capacities.

Flow Network

Flow Network

- Abstraction for material (one commodity!) flowing through the edges
- $G = (V, E)$ directed graph without parallel edges
- distinguished nodes: source s and sink t
- every edge e has a capacity $c(e)$

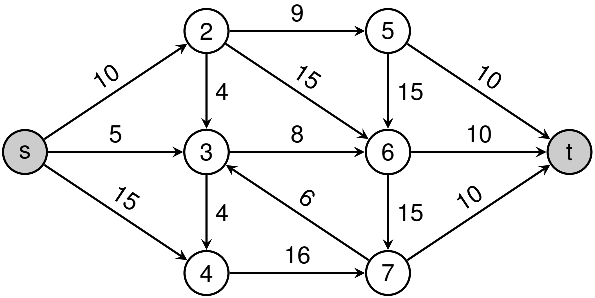


Flow Network

Flow Network

- Abstraction for material (one commodity!) flowing through the edges
- $G = (V, E)$ directed graph without parallel edges
- distinguished nodes: source s and sink t
- every edge e has a capacity $c(e)$

Capacity function $c : V \times V \to \mathbb{R}^+$



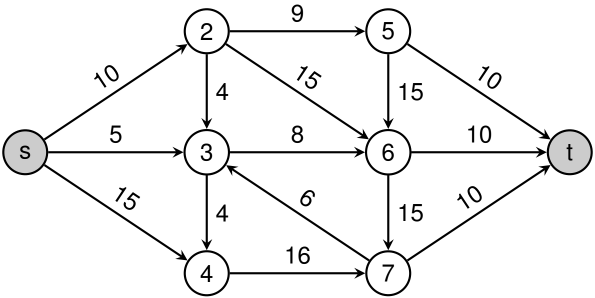
Flow Network

Flow Network

- Abstraction for material (one commodity!) flowing through the edges
- $G = (V, E)$ directed graph without parallel edges
- distinguished nodes: source s and sink t
- every edge e has a capacity $c(e)$

Capacity function $c : V \times V \to \mathbb{R}^+$

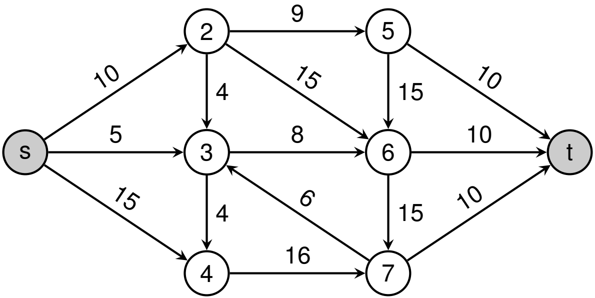
$c(u, v) = 0 \Leftrightarrow (u, v) \notin E$



Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

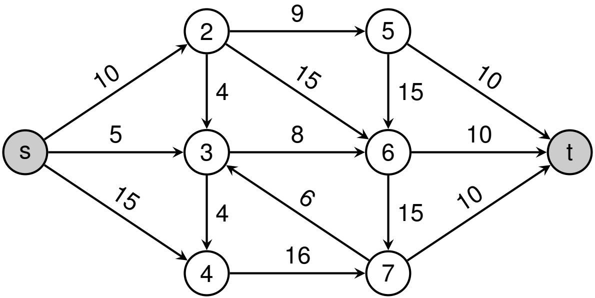


Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$

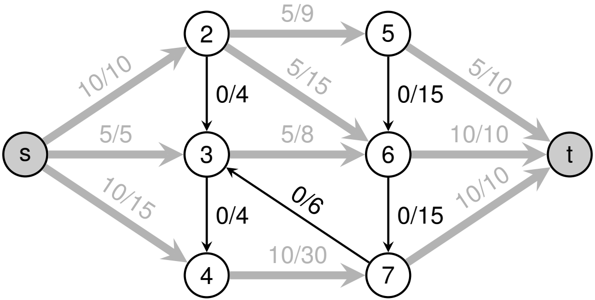


Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$

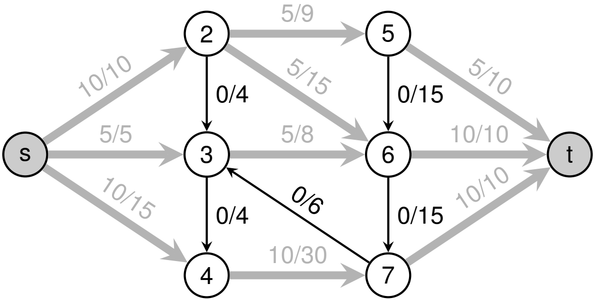


Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u,v) \leq c(u,v)$
- For every $v \in V \setminus \{s,t\}$, $\sum_{(u,v)\in E} f(u,v) = \sum_{(v,u)\in E} f(v,u)$



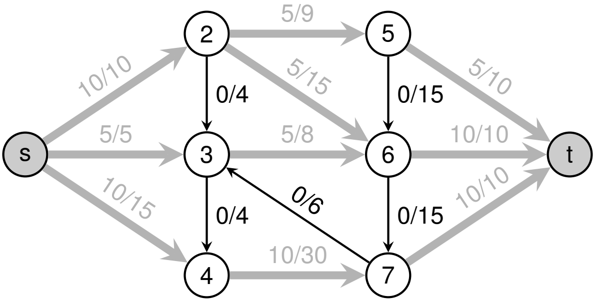
Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$

Flow Conservation



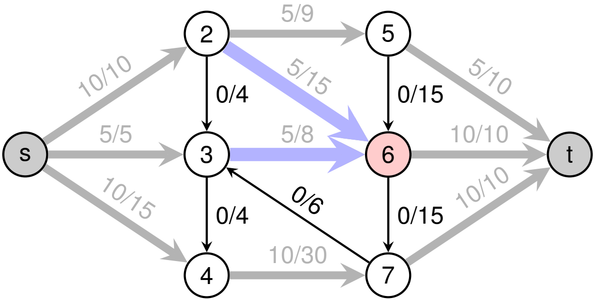
Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$

Flow Conservation



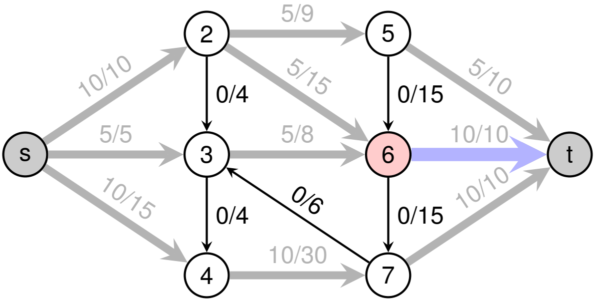
Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$

Flow Conservation



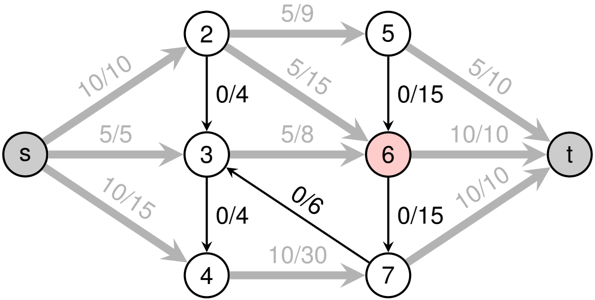
Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$

Flow Conservation

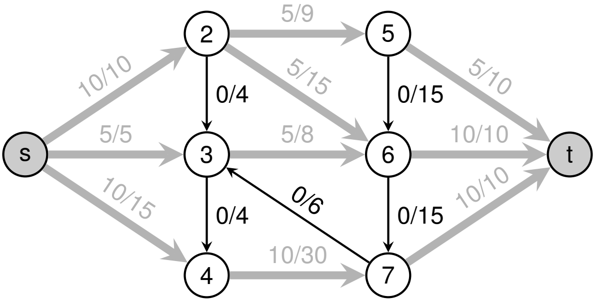


Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$



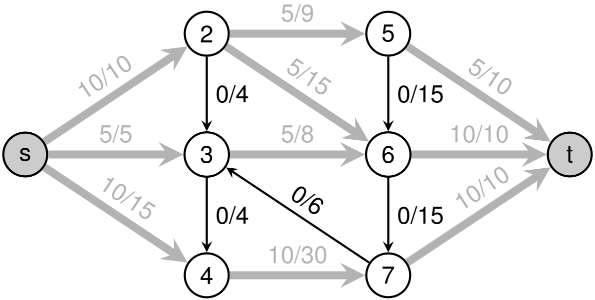
Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,u) \in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v \in V} f(s, v)$



Flow Network

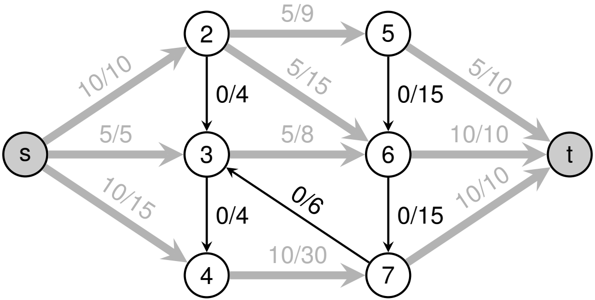
Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v\in V} f(s, v)$

$\sum_{v\in V} f(s, v) = \sum_{v\in V} f(v, t)$



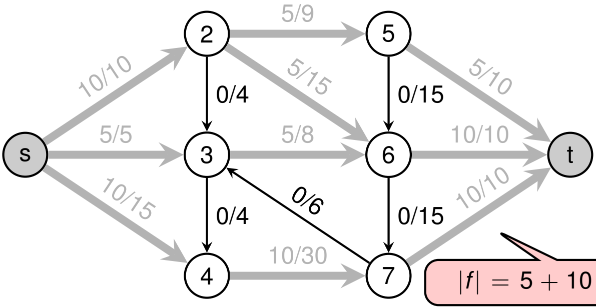
Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v\in V} f(s, v)$



$|f| = 5 + 10 + 10 = 25$

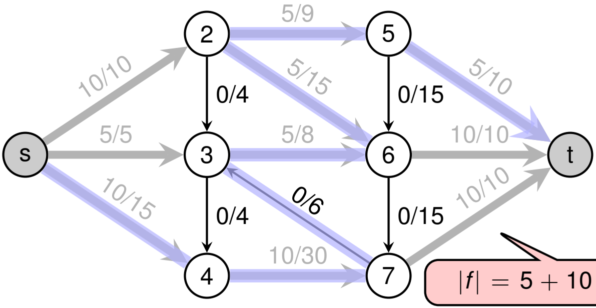
Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \le c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v\in V} f(s, v)$



$|f| = 5 + 10 + 10 = 25$

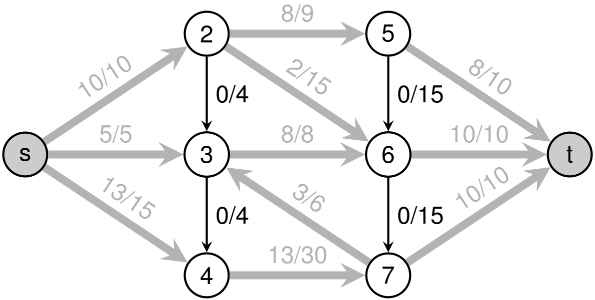
Flow Network

Flow

A flow is a function $f : V \times V \rightarrow \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v\in V} f(s, v)$



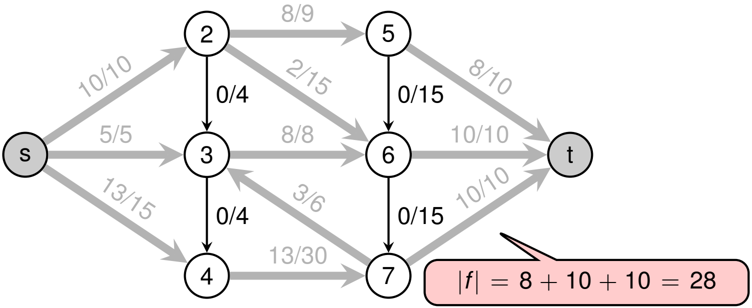
Flow Network

Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v\in V} f(s, v)$



Flow Network

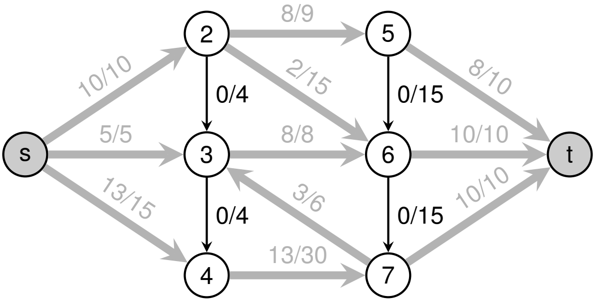
Flow

A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v \in V} f(s, v)$

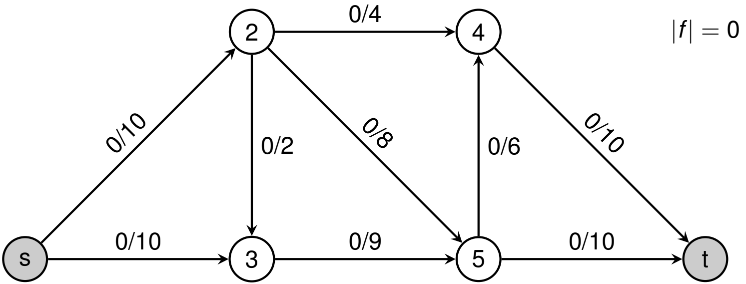
How to find a Maximum Flow?



A First Attempt

Greedy Algorithm

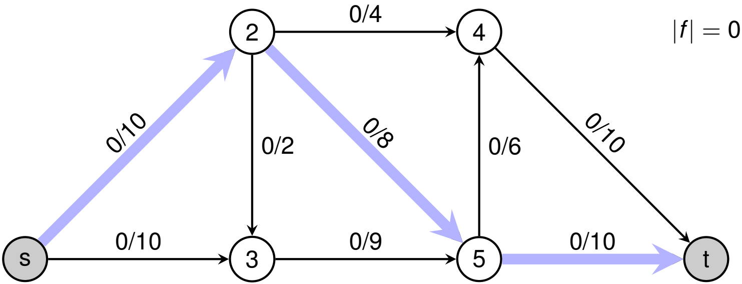
- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s, t)-path p where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along p



A First Attempt

Greedy Algorithm

- Start with $f(u,v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s,t)-path p where each edge $e = (u,v)$ has $f(u,v) < c(u,v)$
  - Augment flow along p

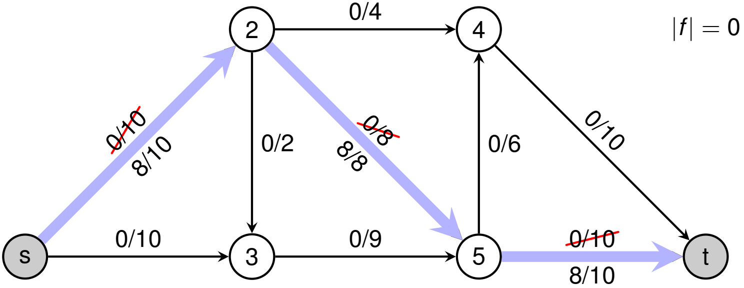


$|f| = 0$

A First Attempt

Greedy Algorithm

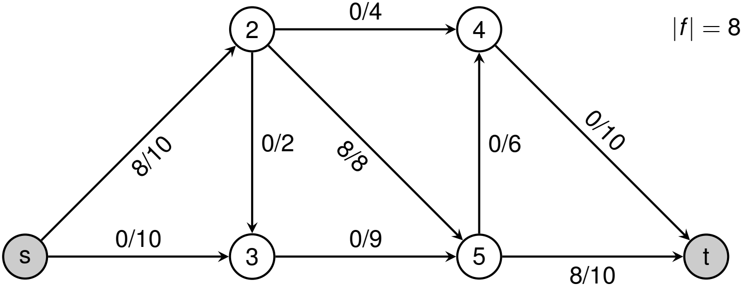
- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s, t)-path p where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along p



A First Attempt

Greedy Algorithm

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s, t)-path p where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along p

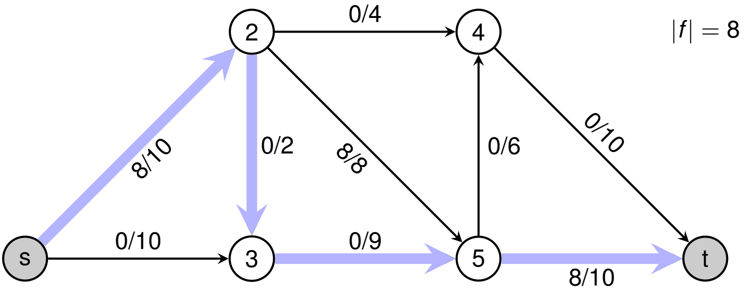


$|f| = 8$

A First Attempt

Greedy Algorithm

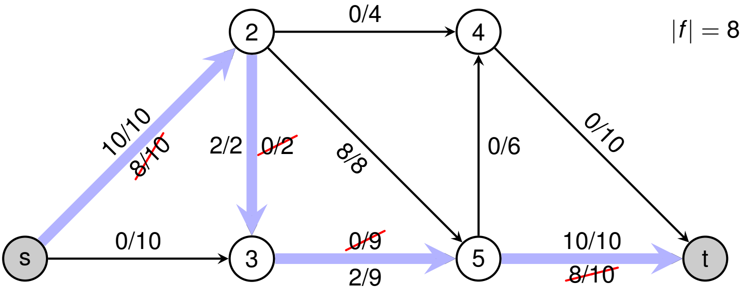
- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s, t)-path p where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along p



A First Attempt

Greedy Algorithm

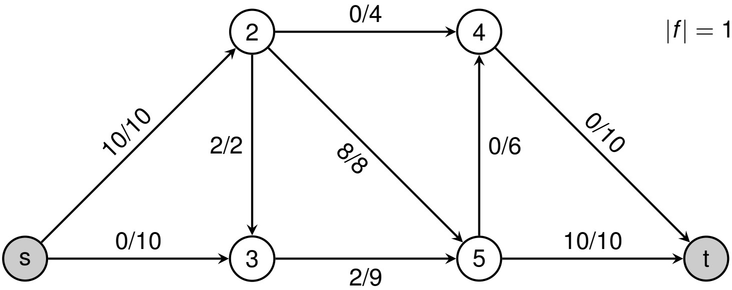
- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s, t)-path p where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along p



A First Attempt

Greedy Algorithm

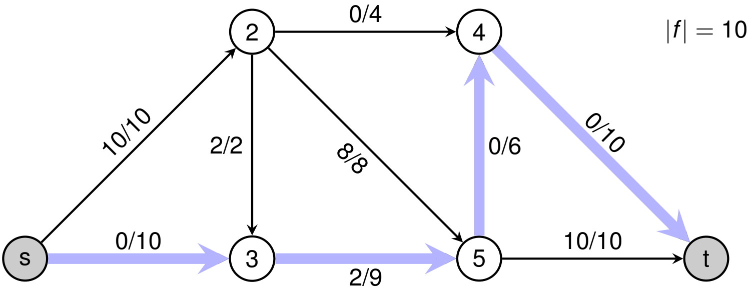
- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s, t)-path p where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along p



A First Attempt

Greedy Algorithm

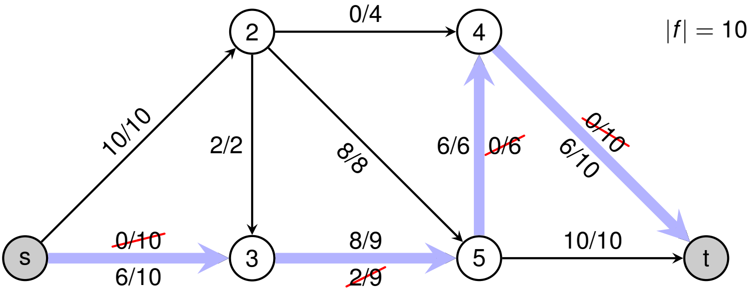
- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s, t)-path p where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along p



A First Attempt

Greedy Algorithm

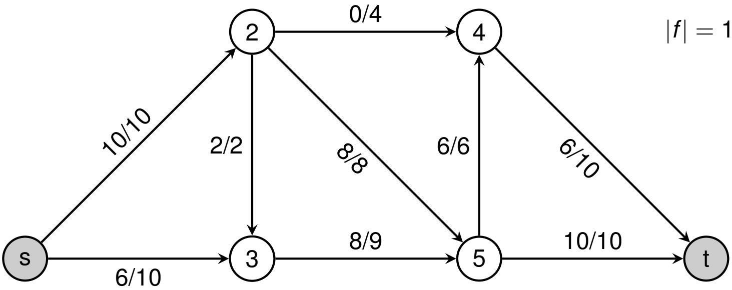
- Start with $f(u,v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s,t)-path p where each edge $e = (u,v)$ has $f(u,v) < c(u,v)$
  - Augment flow along p



A First Attempt

Greedy Algorithm

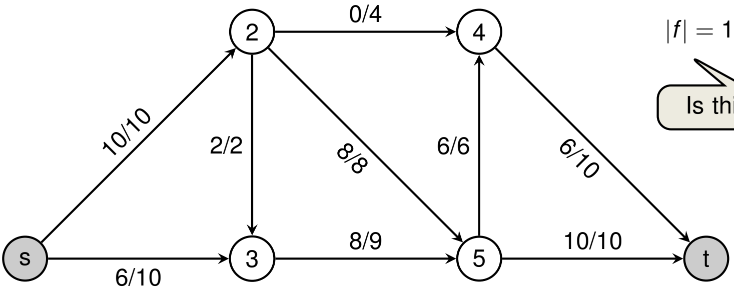
- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s, t)-path p where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along p



A First Attempt

Greedy Algorithm

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s, t)-path p where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along p



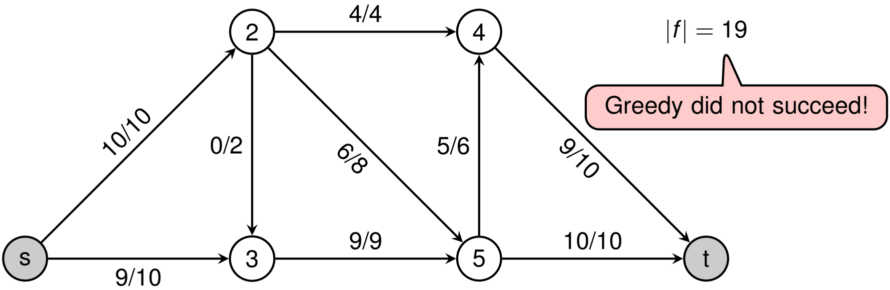
$|f| = 16$

Is this optimal?

A First Attempt

Greedy Algorithm

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a (s, t)-path p where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along p



Outline

Introduction

Ford-Fulkerson

Max-Flow Min-Cut Theorem

Residual Graph

Original Edge

Edge $e = (u, v) \in E$

- flow $f(u, v)$ and capacity $c(u, v)$

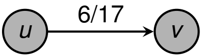
Residual Graph

Original Edge

Edge $e = (u, v) \in E$

- flow $f(u, v)$ and capacity $c(u, v)$

Graph G:



Residual Graph

Original Edge

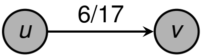
Edge $e = (u, v) \in E$

- flow $f(u, v)$ and capacity $c(u, v)$

Residual Capacity

$$c_f(u, v) = \begin{cases} c(u, v) - f(u, v) & \text{if } (u, v) \in E, \\ f(v, u) & \text{if } (v, u) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Graph G:



Residual Graph

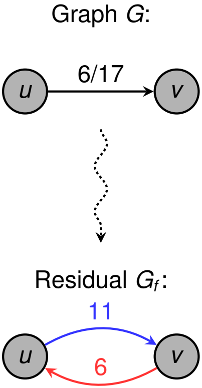
Original Edge

Edge $e = (u, v) \in E$

- flow $f(u, v)$ and capacity $c(u, v)$

Residual Capacity

$$c_f(u, v) = \begin{cases} c(u, v) - f(u, v) & \text{if } (u, v) \in E, \\ f(v, u) & \text{if } (v, u) \in E, \\ 0 & \text{otherwise.} \end{cases}$$



Residual Graph

Original Edge

Edge $e = (u, v) \in E$

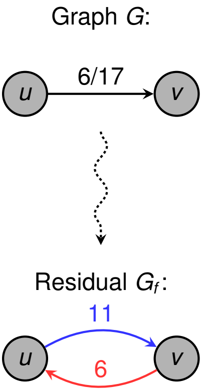
- flow $f(u, v)$ and capacity $c(u, v)$

Residual Capacity

$$c_f(u, v) = \begin{cases} c(u, v) - f(u, v) & \text{if } (u, v) \in E, \\ f(v, u) & \text{if } (v, u) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Residual Graph

- $G_f = (V, E_f, c_f)$, $E_f := \{(u, v) : c_f(u, v) > 0\}$



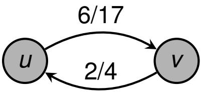
Residual Graph with anti-parallel edges

Original Edge

Edge $e = (u, v) \in E$ (& possibly $e' = (v, u) \in E$)

- flow $f(u, v)$ and capacity $c(u, v)$

Graph G:



Residual Graph with anti-parallel edges

Original Edge

Edge $e = (u, v) \in E$ (& possibly $e' = (v, u) \in E$)

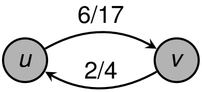
- flow $f(u, v)$ and capacity $c(u, v)$

Residual Capacity

For every pair $(u, v) \in V \times V$,

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

Graph G:



Residual Graph with anti-parallel edges

Original Edge

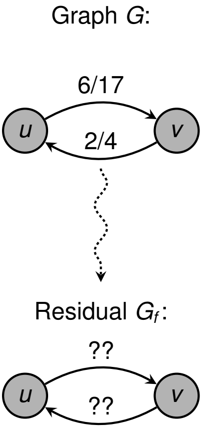
Edge $e = (u, v) \in E$ (& possibly $e' = (v, u) \in E$)

- flow $f(u, v)$ and capacity $c(u, v)$

Residual Capacity

For every pair $(u, v) \in V \times V$,

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



Residual Graph with anti-parallel edges

Original Edge

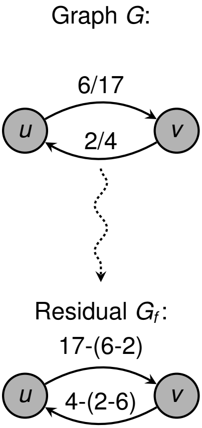
Edge $e = (u, v) \in E$ (& possibly $e' = (v, u) \in E$)

- flow $f(u, v)$ and capacity $c(u, v)$

Residual Capacity

For every pair $(u, v) \in V \times V$,

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



Residual Graph with anti-parallel edges

Original Edge

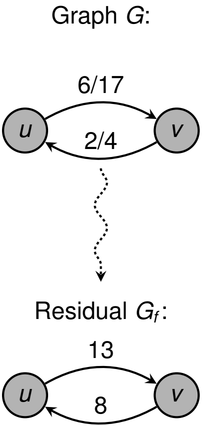
Edge $e = (u, v) \in E$ (& possibly $e' = (v, u) \in E$)

- flow $f(u, v)$ and capacity $c(u, v)$

Residual Capacity

For every pair $(u, v) \in V \times V$,

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



Residual Graph with anti-parallel edges

Original Edge

Edge $e = (u, v) \in E$ (& possibly $e' = (v, u) \in E$)

- flow $f(u, v)$ and capacity $c(u, v)$

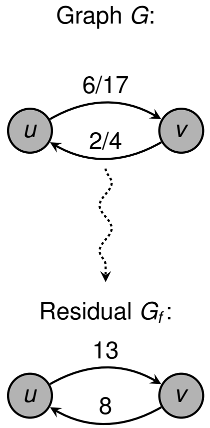
Residual Capacity

For every pair $(u, v) \in V \times V$,

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

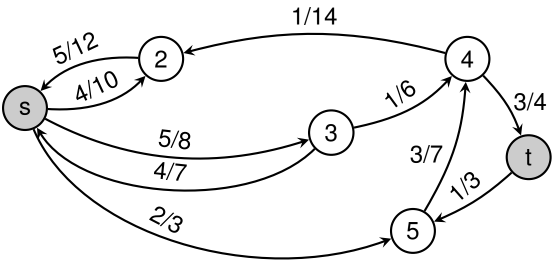
Residual Graph

- $G_f = (V, E_f, c_f)$, $E_f := \{(u, v) \colon c_f(u, v) > 0\}$

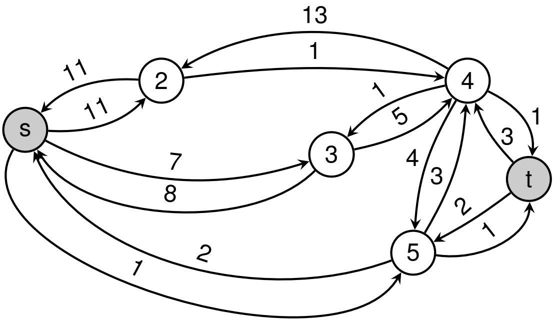


Example of a Residual Graph (Handout)

Flow network G



Residual Graph G_f



The Ford-Fulkerson Method (“Enhanced Greedy”)

```
def fordFulkerson(G)
    initialize flow to 0 on all edges
    while an augmenting path in G_f can be found:
        push as much extra flow as possible through it
```

The Ford-Fulkerson Method ("Enhanced Greedy")

```
def fordFulkerson(G)
    initialize flow to 0 on all edges
    while an augmenting path in G_f can be found:
        push as much extra flow as possible through it
```

Augmenting path: Path from source to sink in G_f

The Ford-Fulkerson Method ("Enhanced Greedy")

```
def fordFulkerson(G)
    initialize flow to 0 on all edges
    while an augmenting path in G_f can be found:
        push as much extra flow as possible through it
```

If f' is a flow in G_f and f a flow in G, then $f + f'$ is a flow in G

The Ford-Fulkerson Method ("Enhanced Greedy")

```
def fordFulkerson(G)
    initialize flow to 0 on all edges
    while an augmenting path in G_f can be found:
        push as much extra flow as possible through it
```

Questions:

- How to find an augmenting path?
- Does this method terminate?
- If it terminates, how good is the solution?

The Ford-Fulkerson Method ("Enhanced Greedy")

```
0: def fordFulkerson(G)
1:     initialize flow to 0 on all edges
2:     while an augmenting path in G_f can be found:
3:         push as much extra flow as possible through it
```

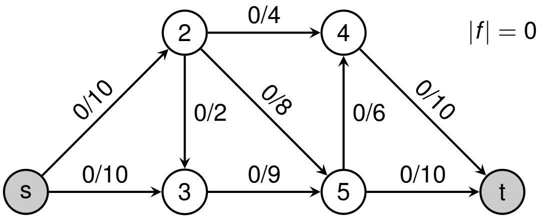
Using BFS or DFS, we can find an augmenting path in $O(V + E)$ time.

Questions:

- How to find an augmenting path?
- Does this method terminate?
- If it terminates, how good is the solution?

Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

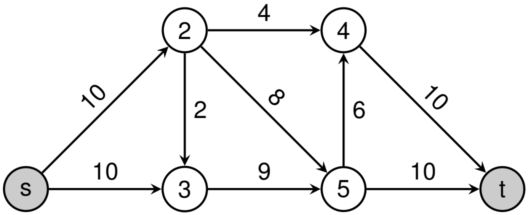
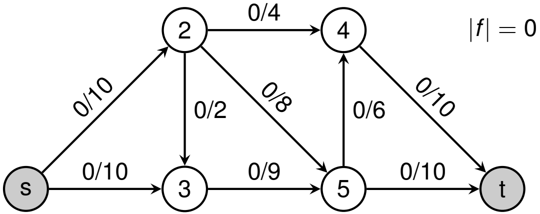


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

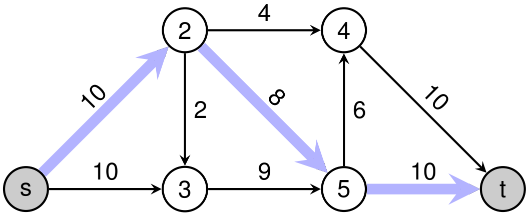
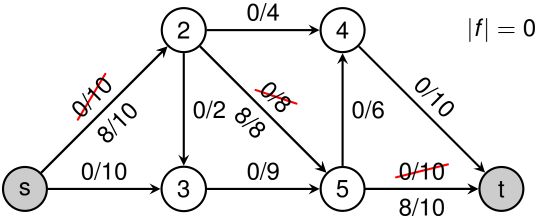


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

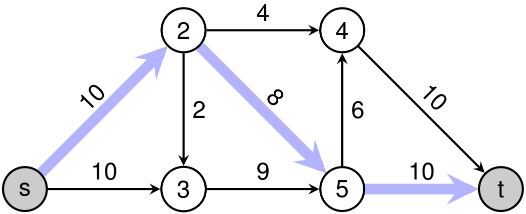
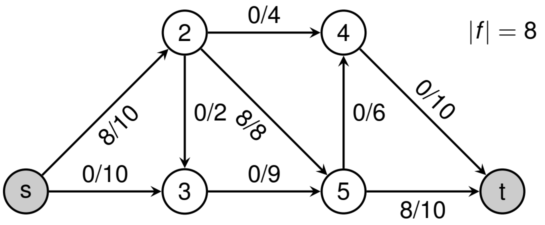


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

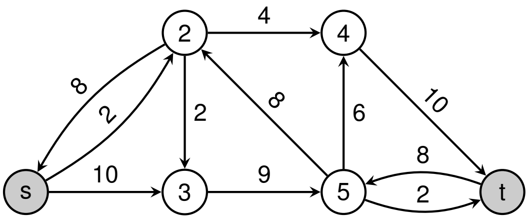
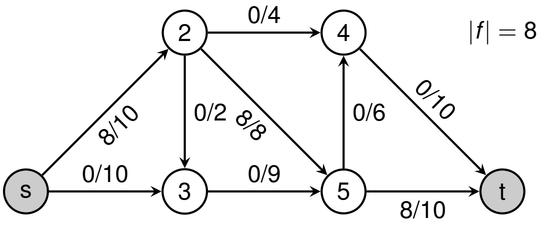


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

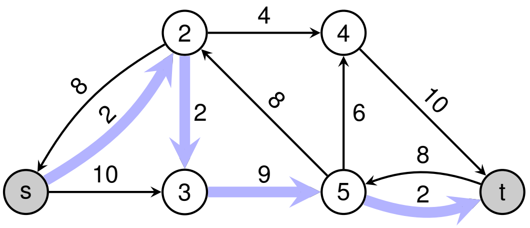
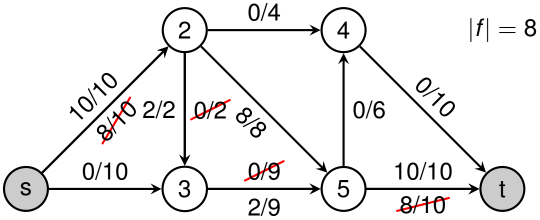


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

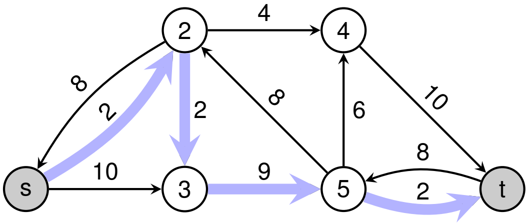
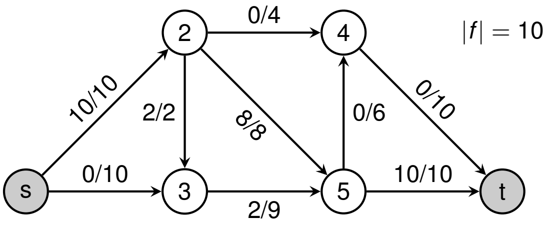


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

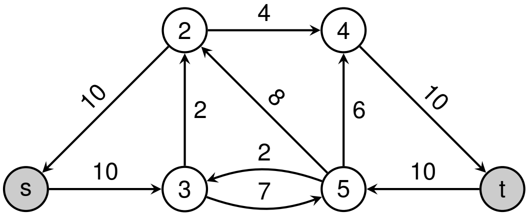
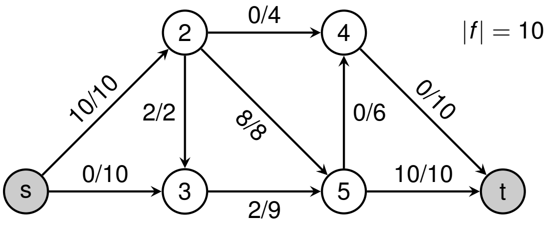


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

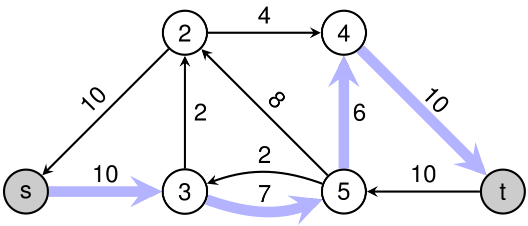
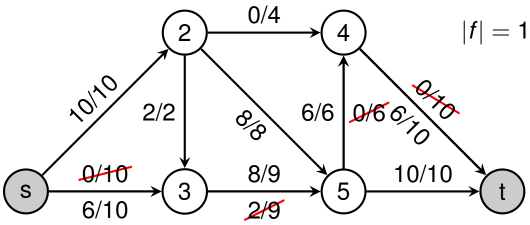


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



$|f| = 10$

Residual Graph $G_f = (V, E_f, c_f)$**:**

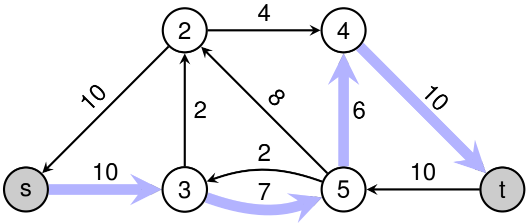
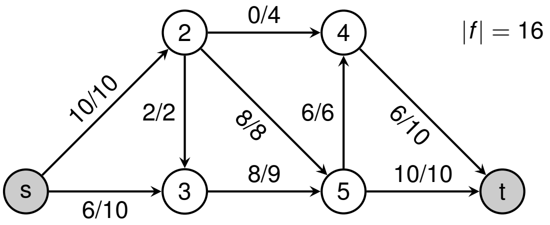


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

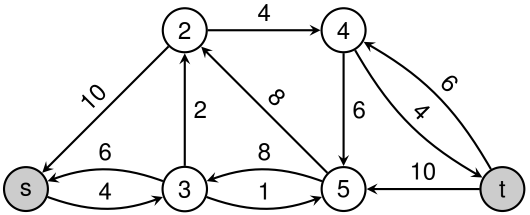
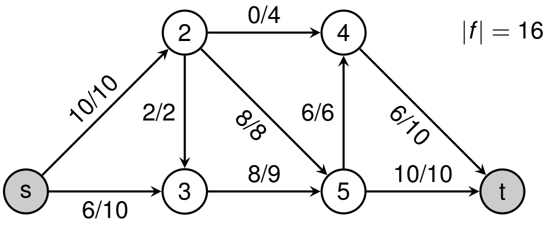


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

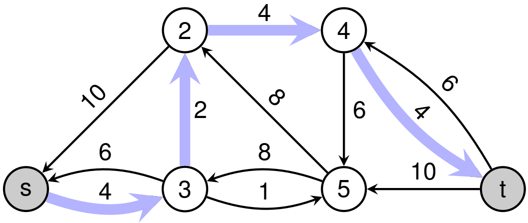
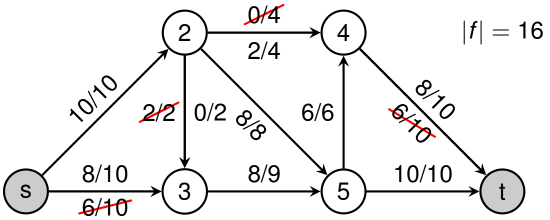


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

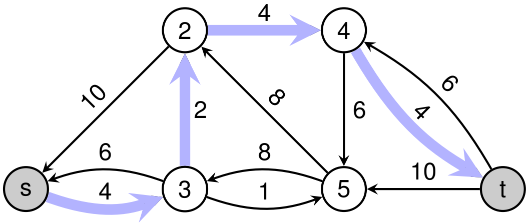
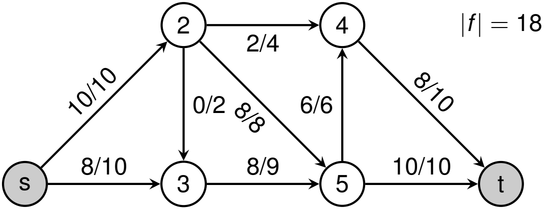


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

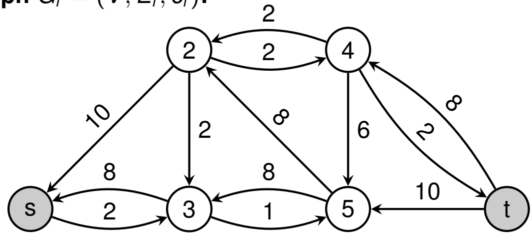
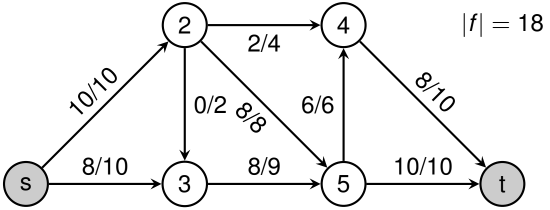


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

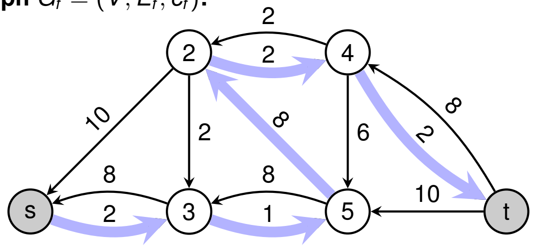
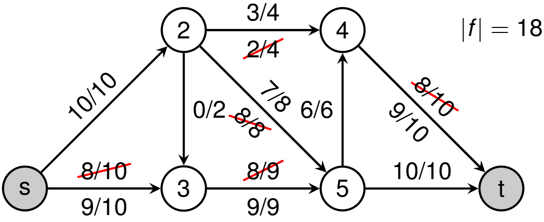


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

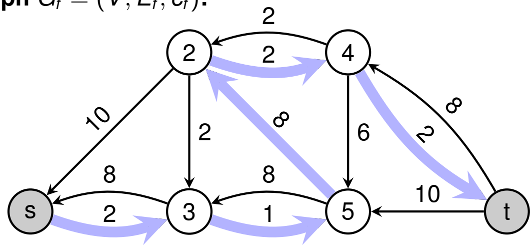
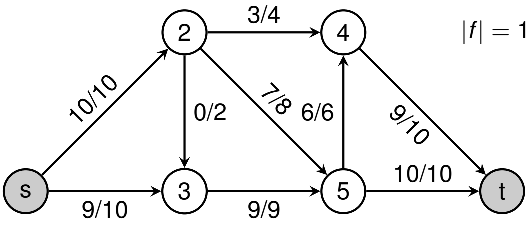


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

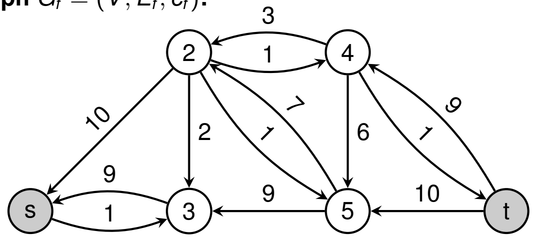
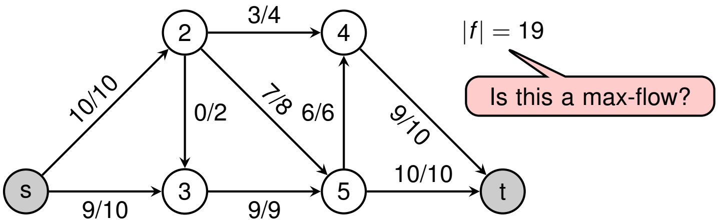


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

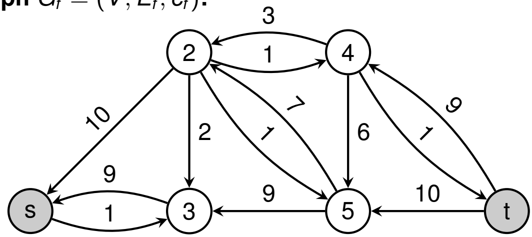
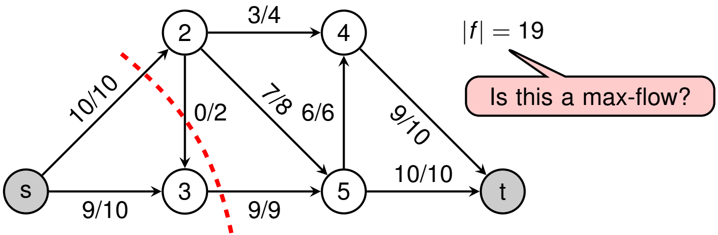


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**

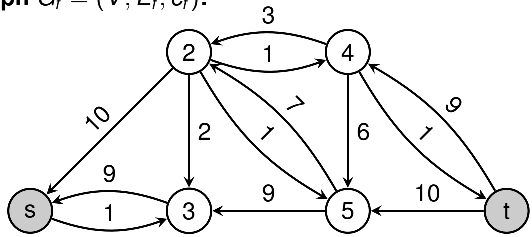
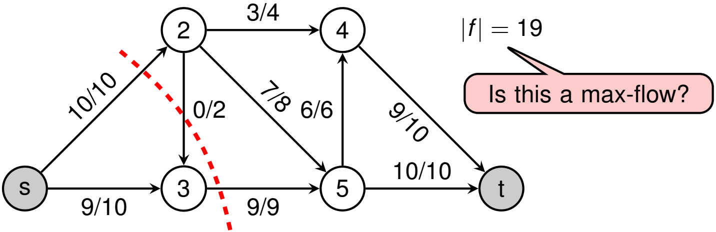
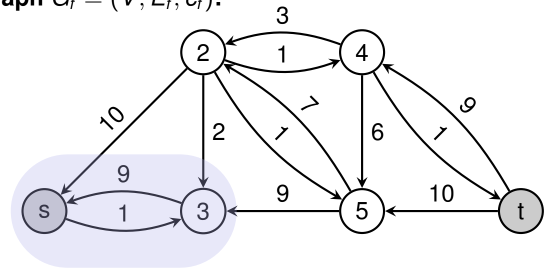


Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$**:**



Residual Graph $G_f = (V, E_f, c_f)$**:**



Outline

Introduction

Ford-Fulkerson

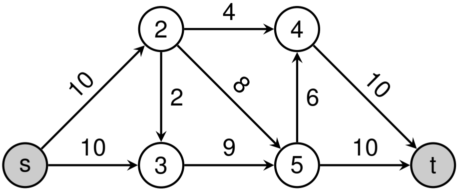
Max-Flow Min-Cut Theorem

From Flows to Cuts

Cut

- A cut (S, T) is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.

Graph $G = (V, E, c)$**:**

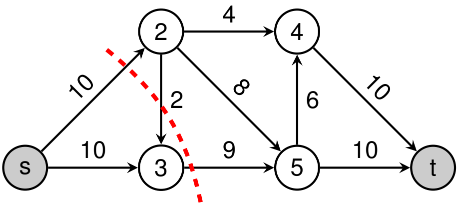


From Flows to Cuts

Cut

- A cut (S,T) is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.

Graph $G = (V, E, c)$**:**



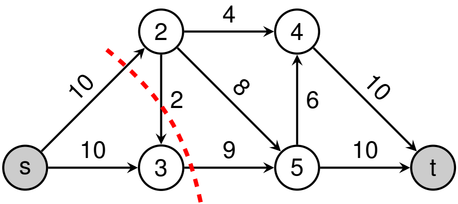
From Flows to Cuts

Cut

- A cut (S, T) is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.
- The capacity of a cut (S, T) is the sum of capacities of the edges from S to T:

$$\mathrm{c}(S,T) = \sum_{u \in S, v \in T} c(u,v) = \sum_{(u,v) \in E(S,T)} c(u,v)$$

Graph $G = (V, E, c)$**:**



$\mathrm{c}(\{s, 3\}, \{2, 4, 5, t\}) =$

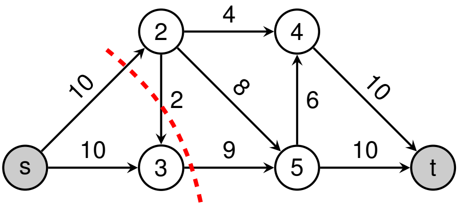
From Flows to Cuts

Cut

- A cut (S, T) is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.
- The capacity of a cut (S, T) is the sum of capacities of the edges from S to T:

$$\mathrm{c}(S,T) = \sum_{u \in S, v \in T} c(u,v) = \sum_{(u,v) \in E(S,T)} c(u,v)$$

Graph $G = (V, E, c)$**:**



$\mathrm{c}(\{s, 3\}, \{2, 4, 5, t\}) = 10 + 9 = 19$

From Flows to Cuts

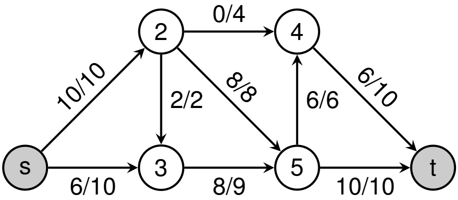
Cut

- A cut (S,T) is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.
- The capacity of a cut (S,T) is the sum of capacities of the edges from S to T:

$$c(S,T) = \sum_{u \in S, v \in T} c(u,v) = \sum_{(u,v) \in E(S,T)} c(u,v)$$

- A mininum cut of a network is a cut whose capacity is minimum over all cuts of the network.

Graph $G = (V, E, c)$**:** $|f| = 16$



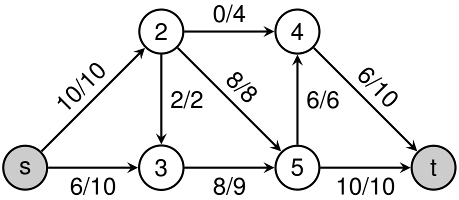
From Flows to Cuts

Flow Value Lemma (Lemma 26.4)

Let f be a flow with source s and sink t, and let (S, T) be any cut of G. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u,v)\in E(S,T)} f(u,v) - \sum_{(v,u)\in E(T,S)} f(v,u).$$

Graph $G = (V, E, c)$**:** $|f| = 16$



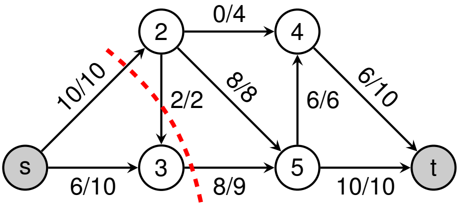
From Flows to Cuts

Flow Value Lemma (Lemma 26.4)

Let f be a flow with source s and sink t, and let (S, T) be any cut of G. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u,v)\in E(S,T)} f(u,v) - \sum_{(v,u)\in E(T,S)} f(v,u).$$

Graph $G = (V, E, c)$: $|f| = 16$



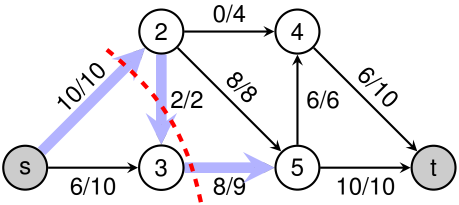
From Flows to Cuts

Flow Value Lemma (Lemma 26.4)

Let f be a flow with source s and sink t, and let (S, T) be any cut of G. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u,v) \in E(S,T)} f(u, v) - \sum_{(v,u) \in E(T,S)} f(v, u).$$

Graph $G = (V, E, c)$**:** $|f| = 16$



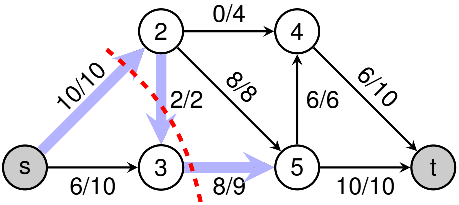
From Flows to Cuts

Flow Value Lemma (Lemma 26.4)

Let f be a flow with source s and sink t, and let (S, T) be any cut of G. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u,v) \in E(S,T)} f(u,v) - \sum_{(v,u) \in E(T,S)} f(v,u).$$

Graph $G = (V, E, c)$: $|f| = 16$



$10 - 2 + 8 = 16$

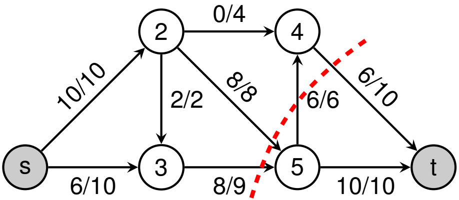
From Flows to Cuts

Flow Value Lemma (Lemma 26.4)

Let f be a flow with source s and sink t, and let (S,T) be any cut of G. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u,v)\in E(S,T)} f(u,v) - \sum_{(v,u)\in E(T,S)} f(v,u).$$

Graph $G = (V,E,c)$**:** $|f| = 16$



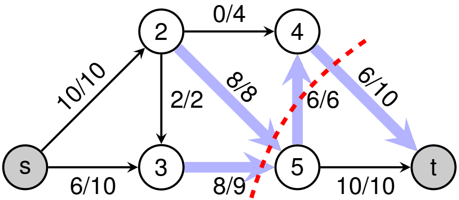
From Flows to Cuts

Flow Value Lemma (Lemma 26.4)

Let f be a flow with source s and sink t, and let (S, T) be any cut of G. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u,v)\in E(S,T)} f(u,v) - \sum_{(v,u)\in E(T,S)} f(v,u).$$

Graph $G = (V, E, c)$**:** $|f| = 16$



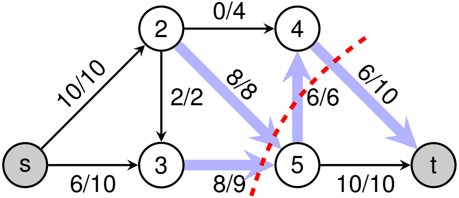
From Flows to Cuts

Flow Value Lemma (Lemma 26.4)

Let f be a flow with source s and sink t, and let (S, T) be any cut of G. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u,v) \in E(S,T)} f(u, v) - \sum_{(v,u) \in E(T,S)} f(v, u).$$

Graph $G = (V, E, c)$: $|f| = 16$



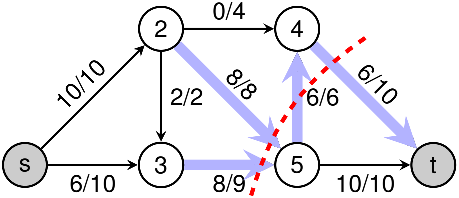
$8 + 8 - 6 + 6 = 16$

From Flows to Cuts

$$|f| = \sum_{(u,v) \in E(S,T)} f(u,v) - \sum_{(v,u) \in E(T,S)} f(v,u).$$

Graph $G = (V, E, c)$**:**

$|f| = 16$



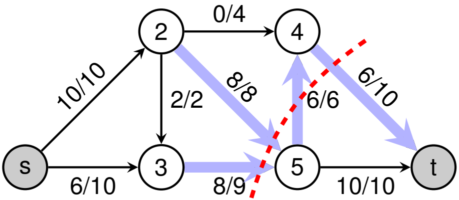
$8 + 8 - 6 + 6 = 16$

From Flows to Cuts

$$|f| = \sum_{(u,v)\in E(S,T)} f(u,v) - \sum_{(v,u)\in E(T,S)} f(v,u).$$

$|f| =$

Graph $G = (V, E, c)$**:** $|f| = 16$



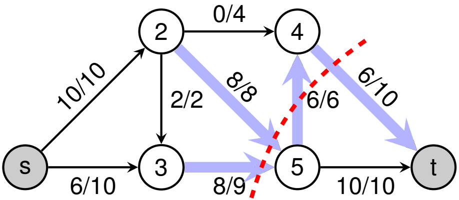
$8 + 8 - 6 + 6 = 16$

From Flows to Cuts

$$|f| = \sum_{(u,v) \in E(S,T)} f(u,v) - \sum_{(v,u) \in E(T,S)} f(v,u).$$

$$|f| = \sum_{w \in V} f(s,w)$$

Graph $G = (V, E, c)$**:** $|f| = 16$



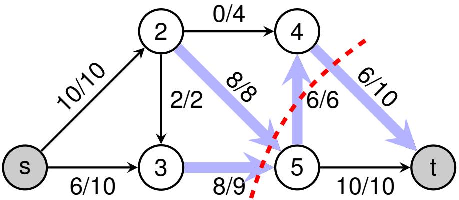
$8 + 8 - 6 + 6 = 16$

From Flows to Cuts

$$|f| = \sum_{(u,v)\in E(S,T)} f(u,v) - \sum_{(v,u)\in E(T,S)} f(v,u).$$

$$|f| = \sum_{w\in V} f(s,w) = \sum_{u\in S}\left(\sum_{(u,w)\in E} f(u,w) - \sum_{(w,u)\in E} f(w,u)\right)$$

Graph $G = (V, E, c)$**:** $|f| = 16$



$8 + 8 - 6 + 6 = 16$

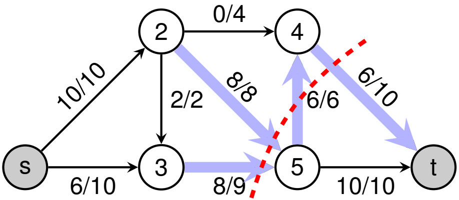
From Flows to Cuts

$$|f| = \sum_{(u,v)\in E(S,T)} f(u,v) - \sum_{(v,u)\in E(T,S)} f(v,u).$$

$$|f| = \sum_{w\in V} f(s,w) = \sum_{u\in S}\left(\sum_{(u,w)\in E} f(u,w) - \sum_{(w,u)\in E} f(w,u)\right)$$

$$= \sum_{(u,v)\in E(S,T)} f(u,v) - \sum_{(v,u)\in E(T,S)} f(v,u) \qquad \square$$

Graph $G = (V, E, c)$**:** $|f| = 16$



$8 + 8 - 6 + 6 = 16$