CSE 417 Autumn 2025

# Lecture 21: Max Flow

Nathan Brunelle

# Origins of Max Flow and Min Cut Problems

Max Flow problem formulation:

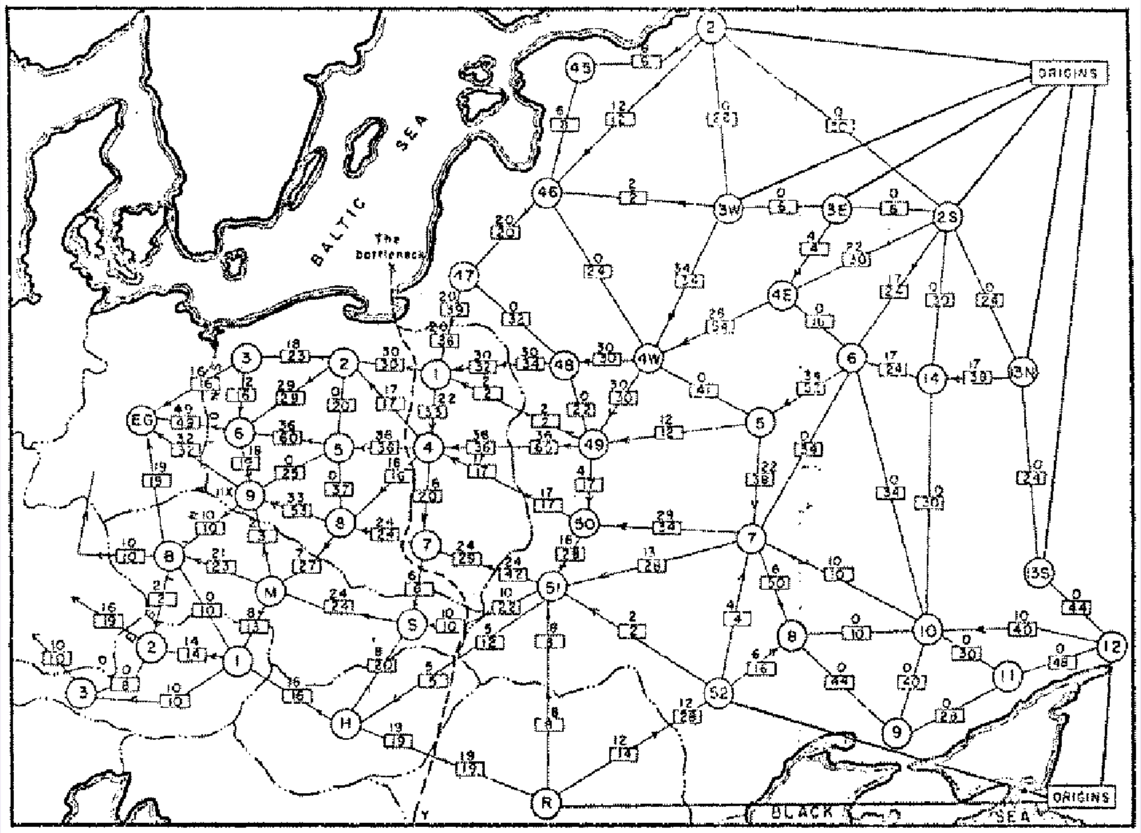
[Tolstoy 1930] Rail transportation planning for the Soviet Union

Min Cut problem formulation:

Cold War:  US military planners want to find a way to cripple Soviet supply routes

[Harris 1954] Secret RAND corp report for US Air Force

[Ford-Fulkerson 1955] Problems are equivalent




Reference:  *On the history of the transportation and maximum flow problems*.
Alexander Schrijver in Math Programming, 91: 3, 2002.

# Flow Network

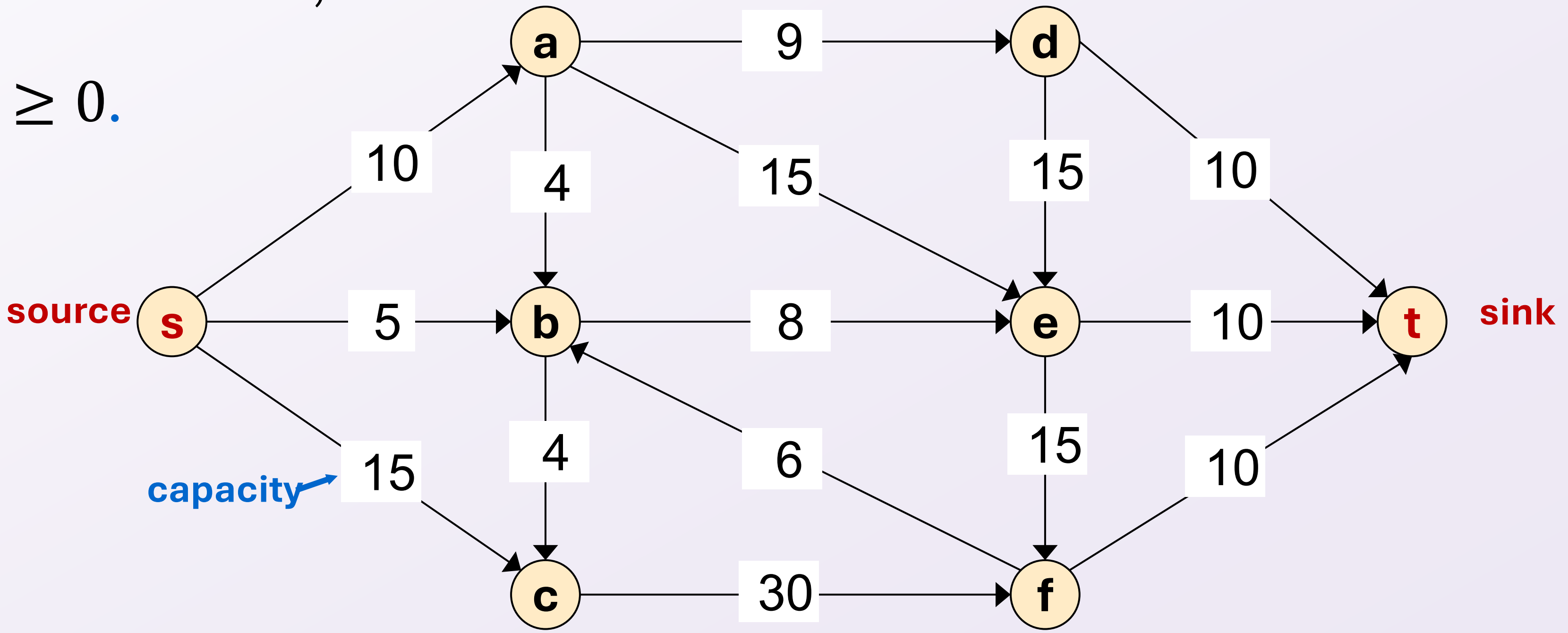
## Flow network:

Abstraction for material *flowing* through the edges.

$G = (V, E)$ directed graph, no parallel edges.

Two distinguished nodes: $s$ = source, $t$ = sink.

$c(e)$ = capacity of edge $e \geq 0$.



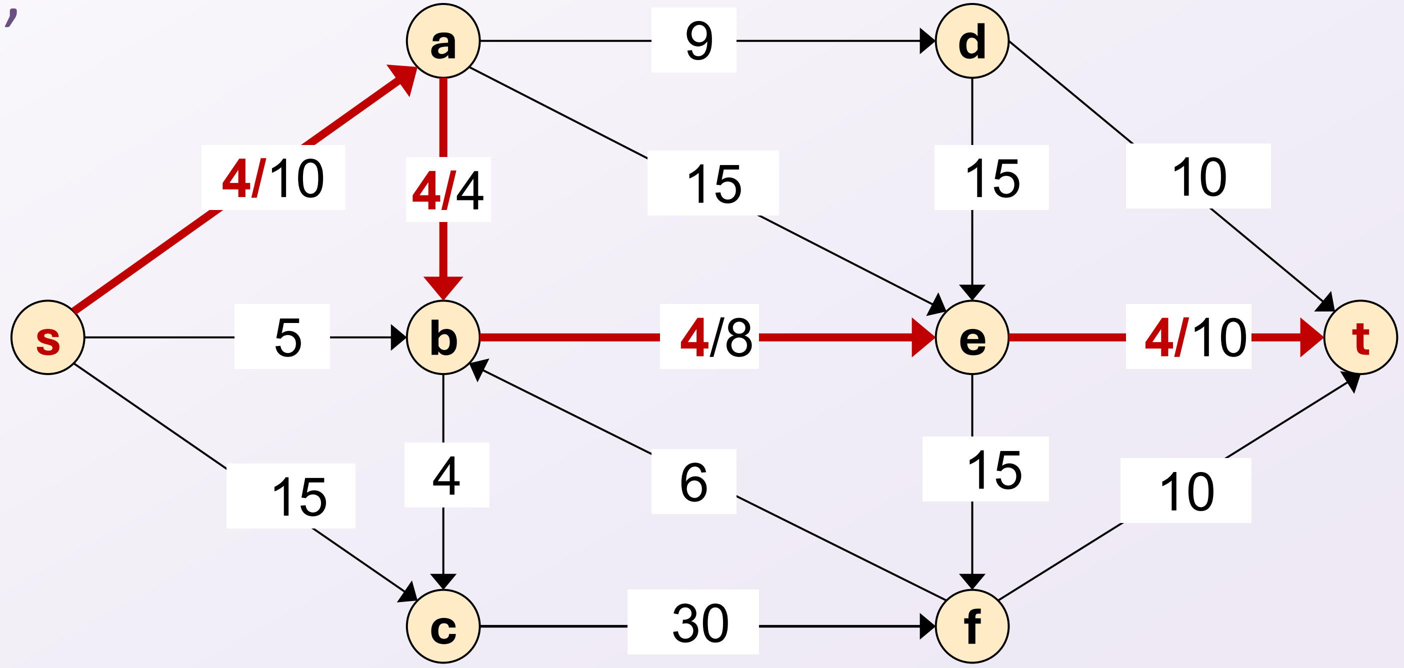
# Flow Graph

**Defn:** An $s$-$t$ flow in a flow network is a function $f\colon\ E \to \mathbb{R}$ that satisfies:

- For each $e \in E$: $0 \leq\ f(e) \leq\ c(e)$ [capacity constraints]
- For each $v \in V - \{s, t\}$ : $\displaystyle\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$ [flow conservation]

**Defn:** The value of flow $f$,

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




Only show non-zero values of $f$

value = **4**

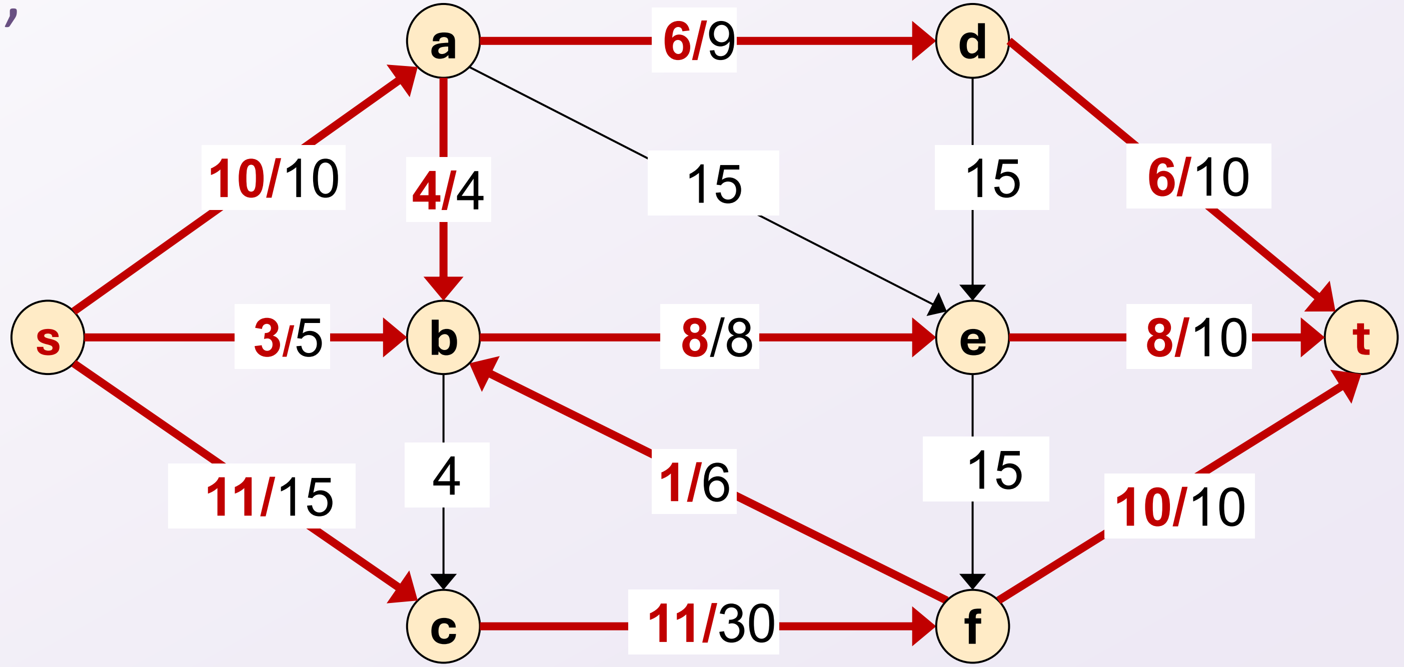
# Another Flow Graph

**Defn:** An $s$-$t$ flow in a flow network is a function $f\colon E \to \mathbb{R}$ that satisfies:

- For each $e \in E$: $0 \le f(e) \le c(e)$ [capacity constraints]
- For each $v \in V - \{s,t\}$ : $\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$ [flow conservation]

**Defn:** The value of flow $f$,

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



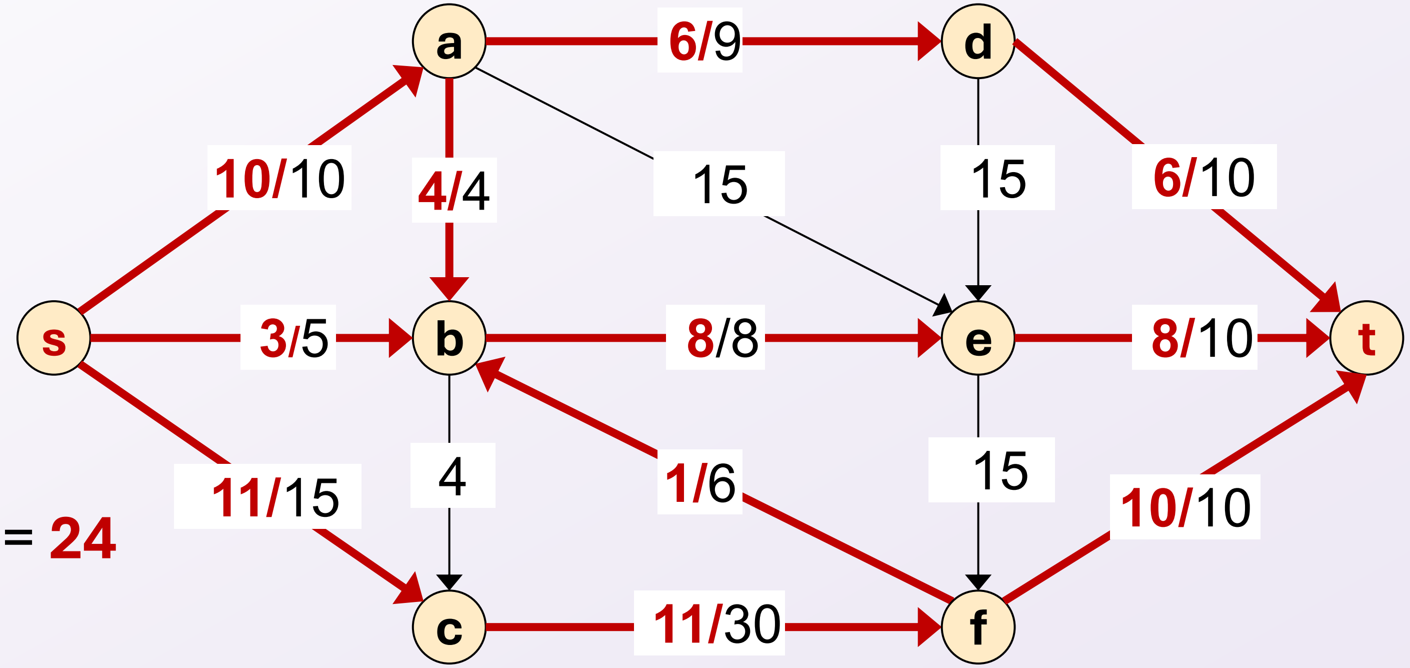

Only show non-zero values of $f$

value = **24**

# Maximum Flow Problem

**Given:** a flow network

**Find:** an $s$-$t$ flow of maximum value



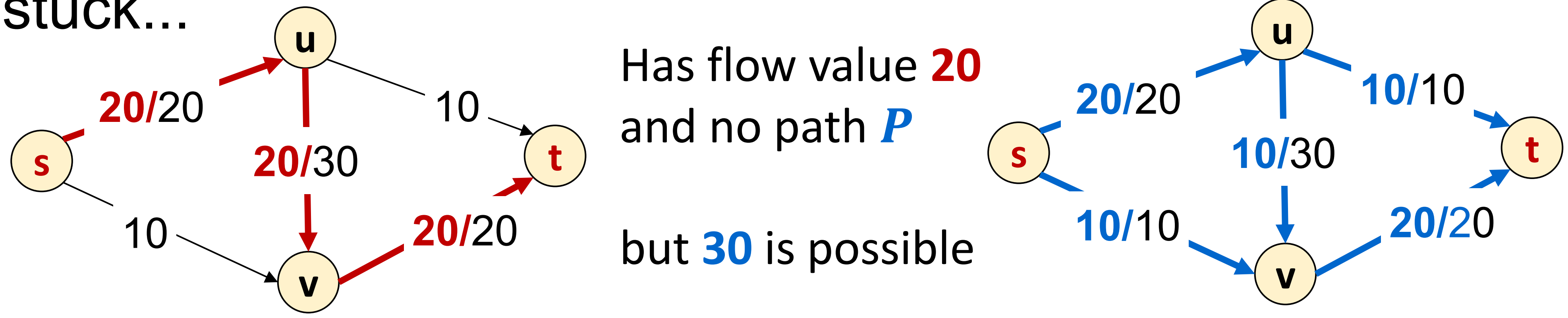

value = **24**

# Attempt at a Greedy Solution

What about the following greedy algorithm?

- Start with $f(e) = 0$ for all edges $e \in E$.
- While there is an $s$-$t$ path $P$ where each edge has $f(e) < c(e)$.
    - "Augment" flow along $P$; that is:
        - Let $\alpha = \min_{e \in P}(c(e) - f(e))$
        - Add $\alpha$ to flow on every edge $e$ along path $P$. (Adds $\alpha$ to $v(f)$.)

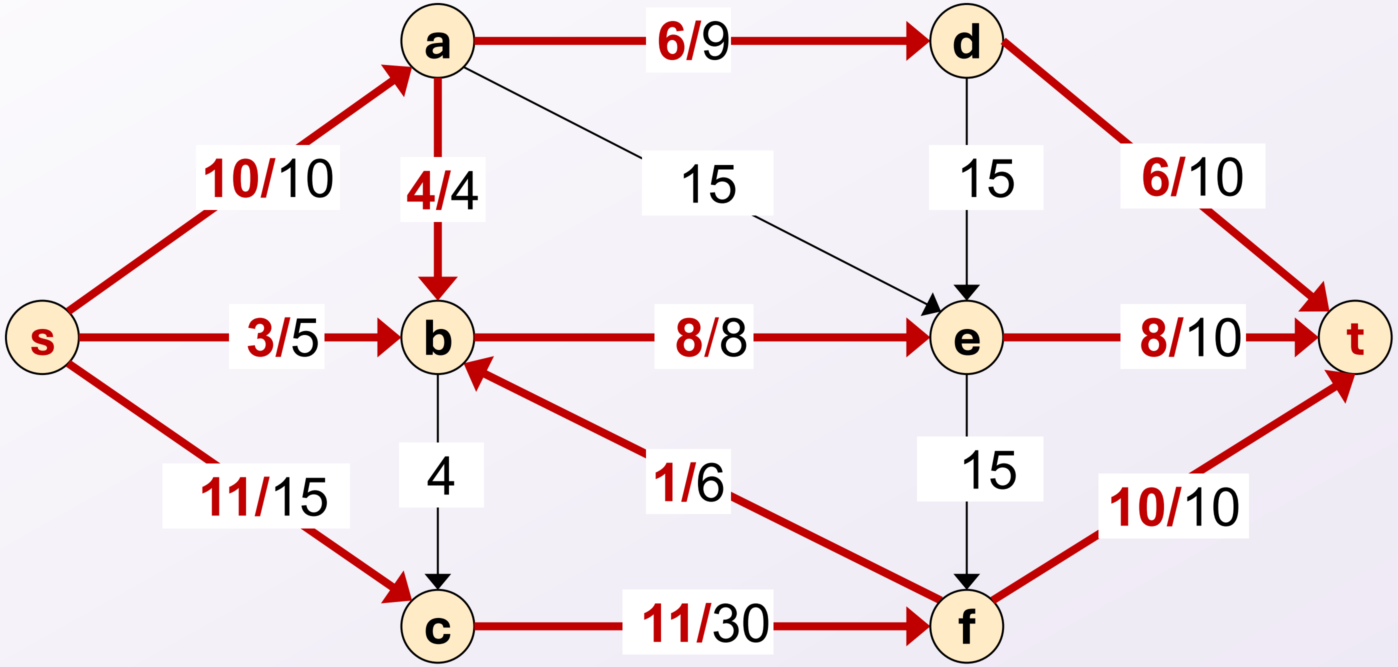
Can get stuck...



# Another "Stuck" Example

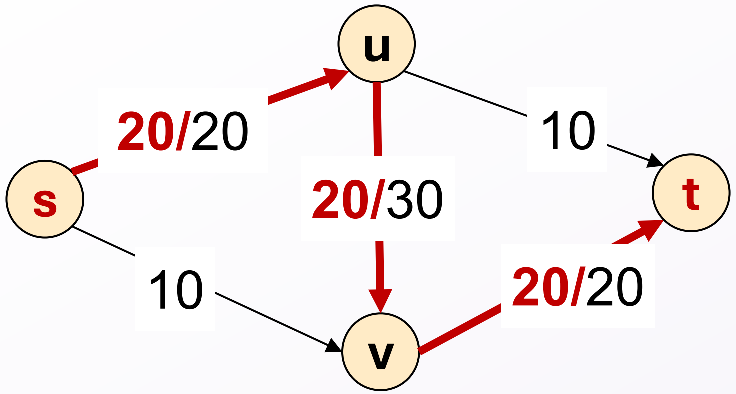
On every $s$-$t$ path there is some edge with $f(e) = c(e)$:

Value of flow = **24**




**Next idea:** Ford-Fulkerson Algorithm, which applies greedy ideas to a "residual graph" that lets us reverse prior flow decisions from the basic greedy approach to get optimal results!

# Greed Revisited: Residual Graph & Augmenting Paths



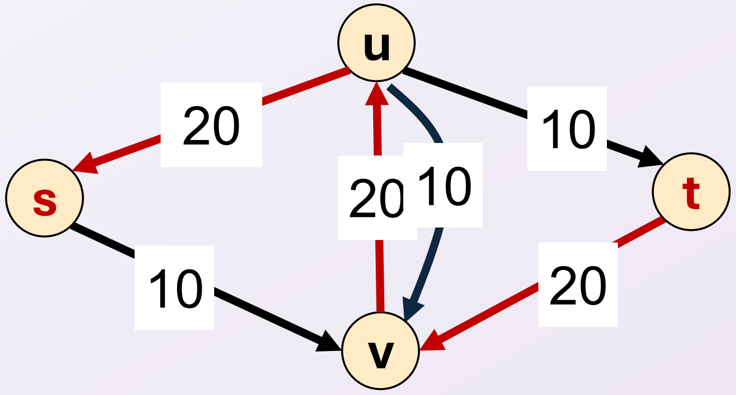

To get more flow from **s** to **t** we need to reduce the flow from **u** to **v**

To conserve flow in and out of **u** we need to increase the flow from **u** to **t** by that same amount.

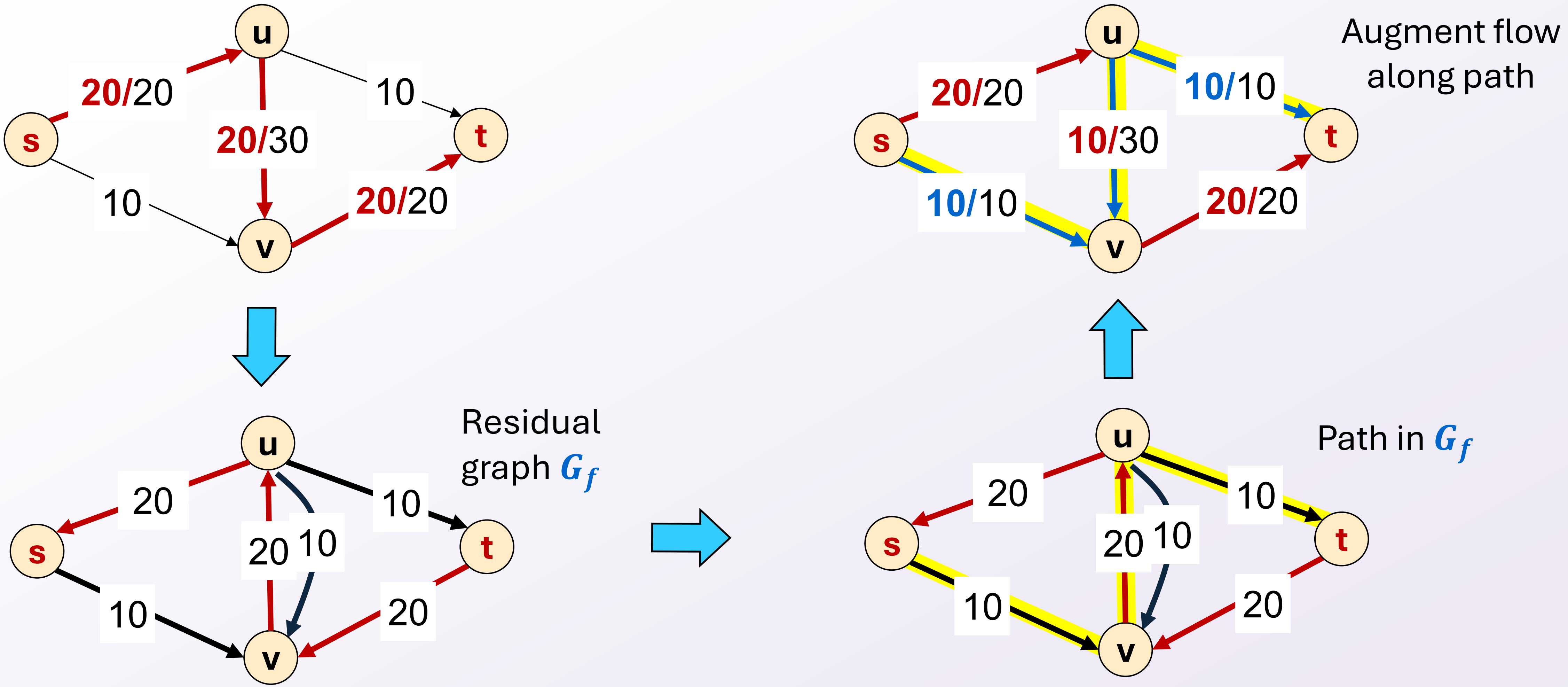
Suppose that we took this flow $f$ as a baseline, what changes could each edge handle?

- We could add up to 10 units along **sv** or **ut** or **uv**
- We could reduce by up to 20 units from **su** or **uv** or **vt**

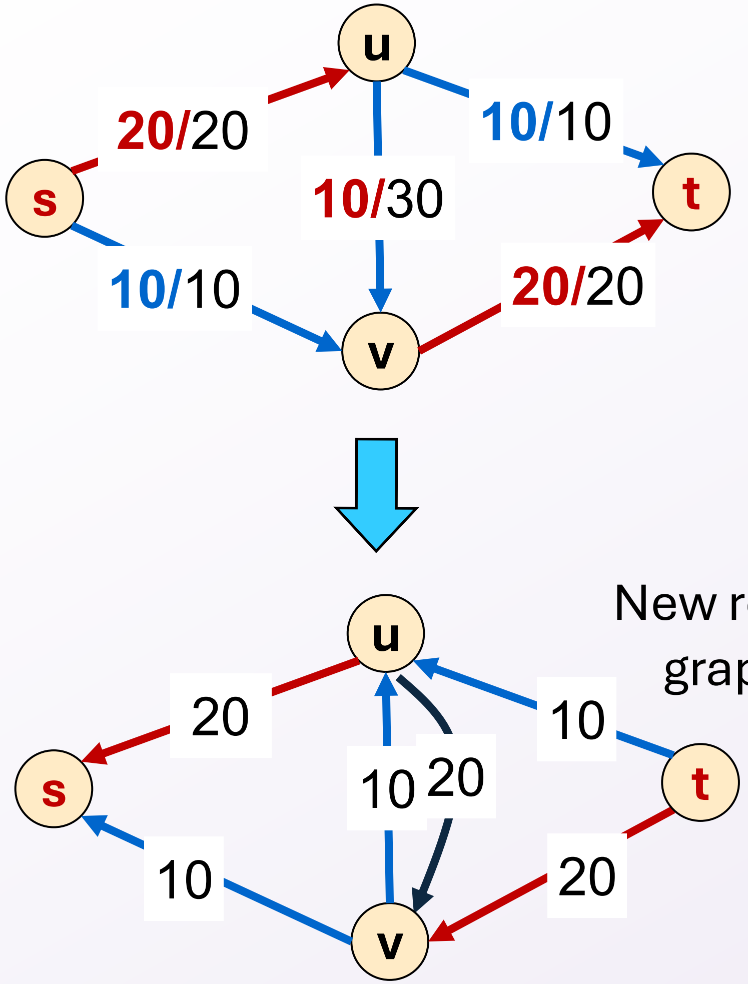
This gives us a residual graph $G_f$ of possible changes where we draw reducing as "sending back".



# Greed Revisited: Residual Graph & Augmenting Paths



# Greed Revisited: Residual Graph & Augmenting Paths




New residual graph $G_f$

No path can even leave $s$!

# Residual Graphs

An alternative way to represent a flow network

Represents the net available flow between two nodes

Original edge: $e = (u, v) \in E$.

Flow $f(e)$, capacity $c(e)$.

Residual edges of two kinds:

Forward: $e = (u, v)$ with capacity $c_f(e) = c(e) - f(e)$

- Amount of extra flow we can add along $e$

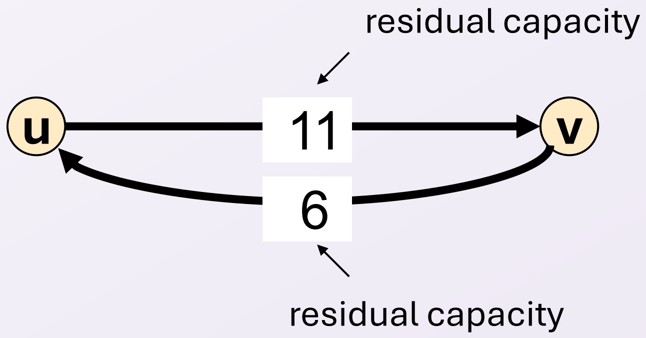
Backward: $e^{\mathrm{R}} = (v, u)$ with capacity $c_f(e) = f(e)$

- Amount we can reduce/undo flow along $e$

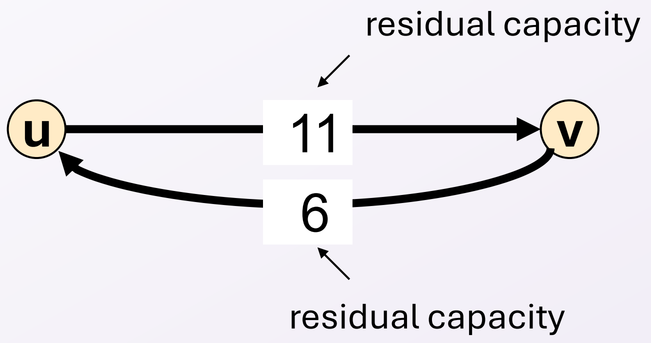
Residual graph: $G_f = (V, E_f)$.

Residual edges with residual capacity $c_f(e) > 0$.

$E_f = \{e : f(e) < c(e)\} \cup \{e^{\mathrm{R}} : f(e) > 0\}$.

# Residual Graphs and Augmenting Paths




Residual edges of two kinds:

Forward: $e = (u, v)$ with capacity $c_f(e) = c(e) - f(e)$

- Amount of extra flow we can add along $e$

Backward: $e^{\mathrm{R}} = (v, u)$ with capacity $c_f(e) = f(e)$

- Amount we can reduce/undo flow along $e$

Residual graph: $G_f = (V, E_f)$.

Residual edges with residual capacity $c_f(e) > 0$.

$E_f = \{e : f(e) < c(e)\} \cup \{e^{\mathrm{R}} : f(e) > 0\}$.

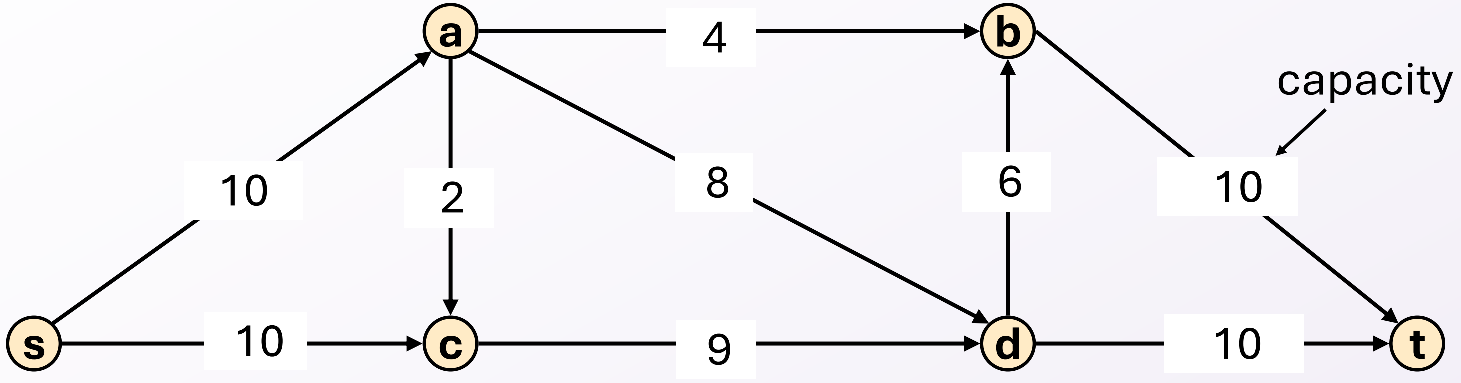
Augmenting Path: Any $s$-$t$ path $P$ in $G_f$. Let **bottleneck**$(P) = \min_{e \in P} c_f(e)$.

**Ford-Fulkerson idea:** Repeat "find an augmenting path $P$ and increase flow by **bottleneck**$(P)$" until none left.

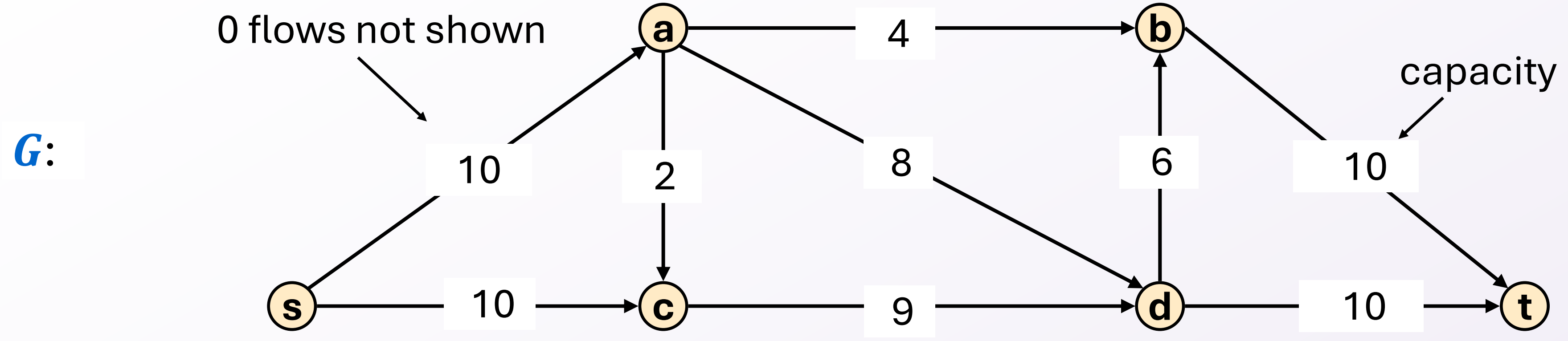

13

# Ford-Fulkerson Algorithm

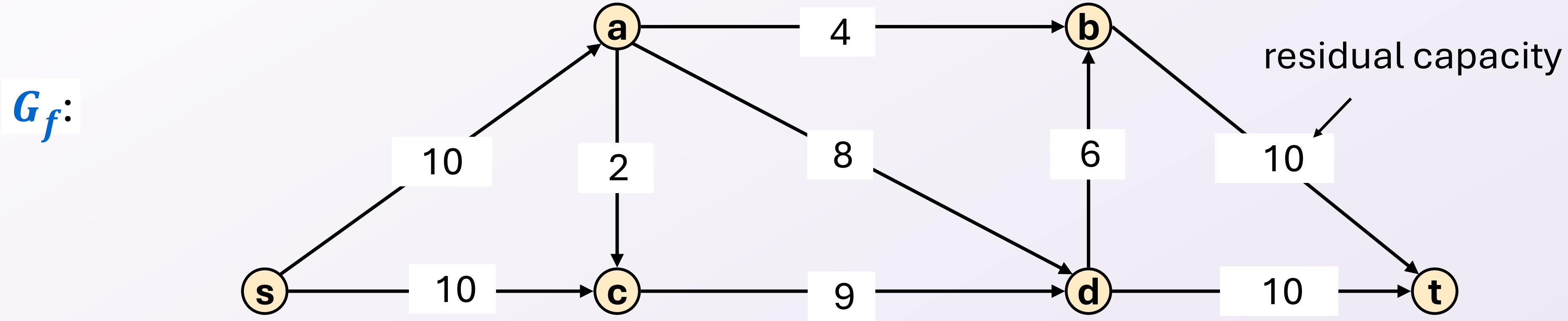
$G$:



# Ford-Fulkerson Algorithm

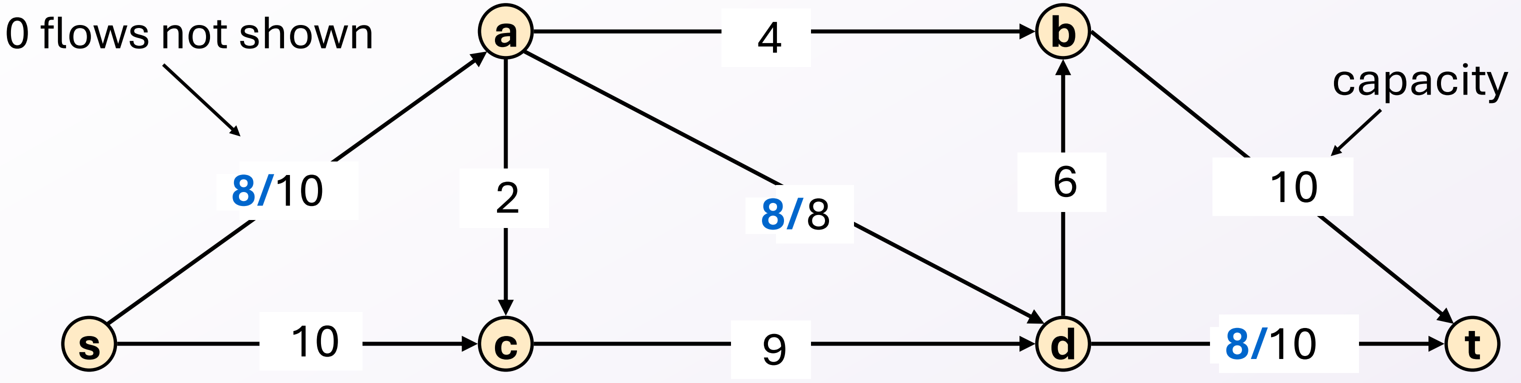



Flow value = **0**



# Ford-Fulkerson Algorithm

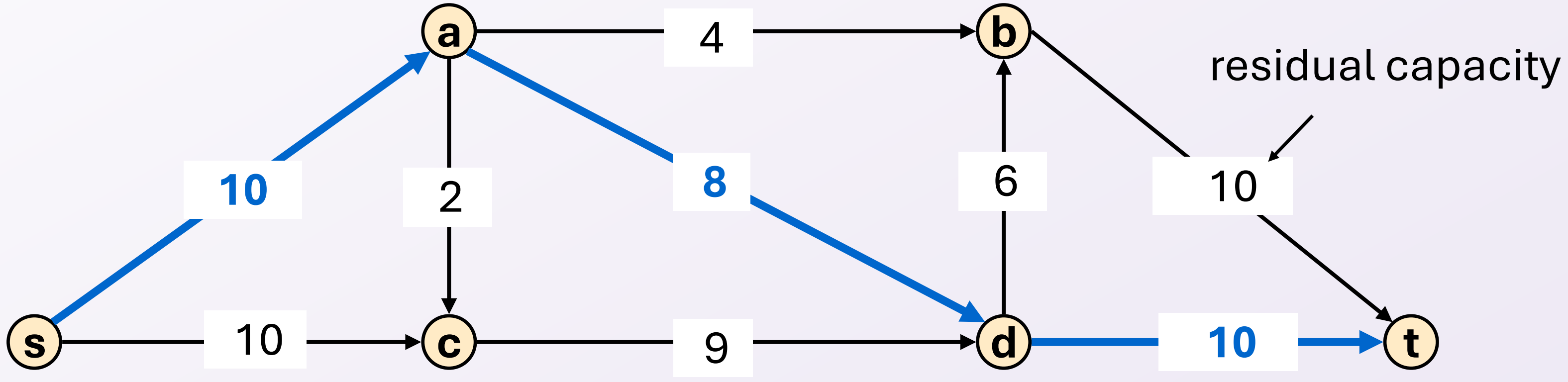
$G$:




Flow value = 0

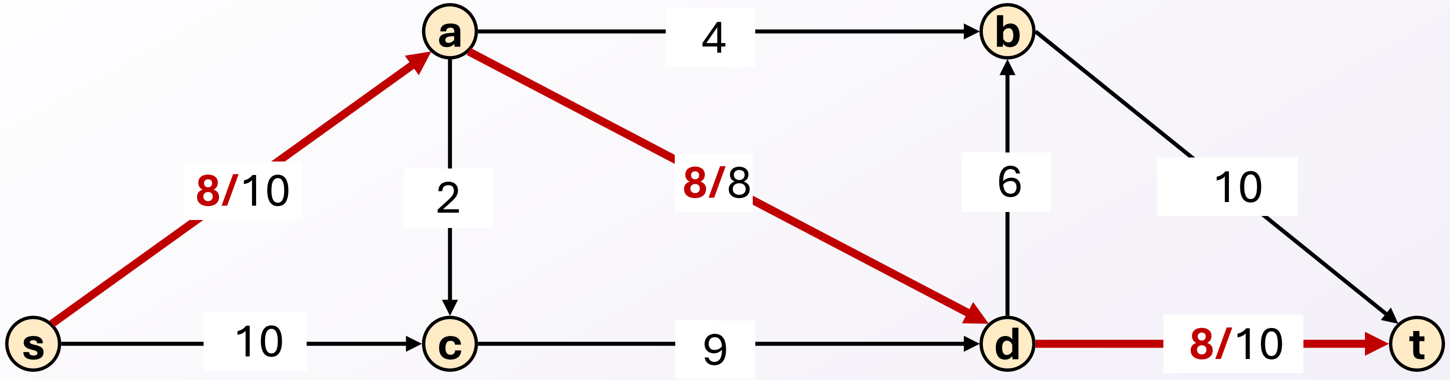
+8=8

$G_f$:



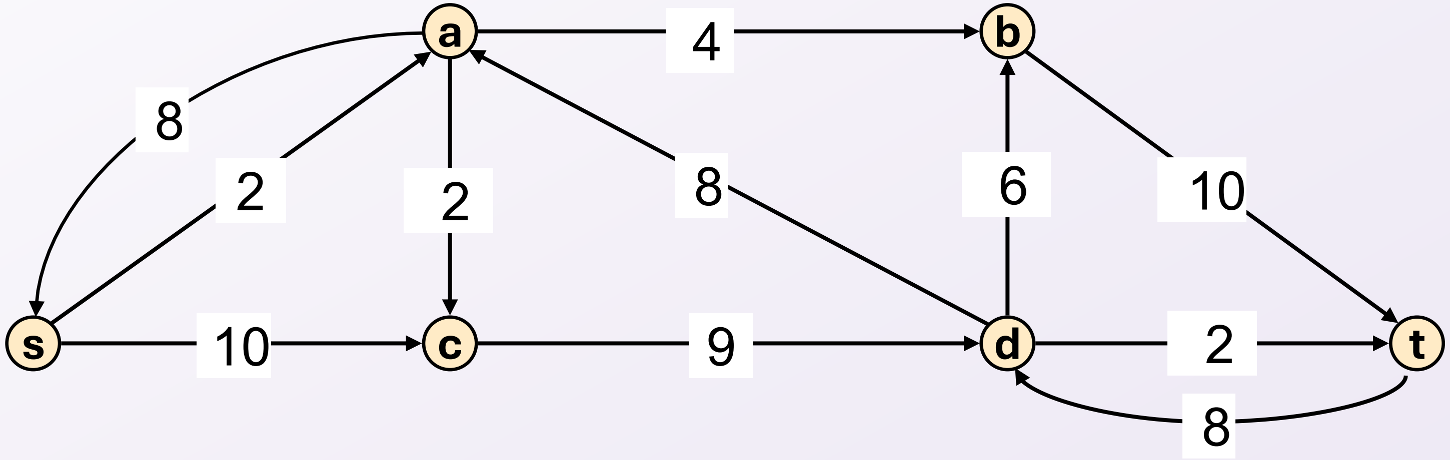
# Ford-Fulkerson Algorithm

$G$:



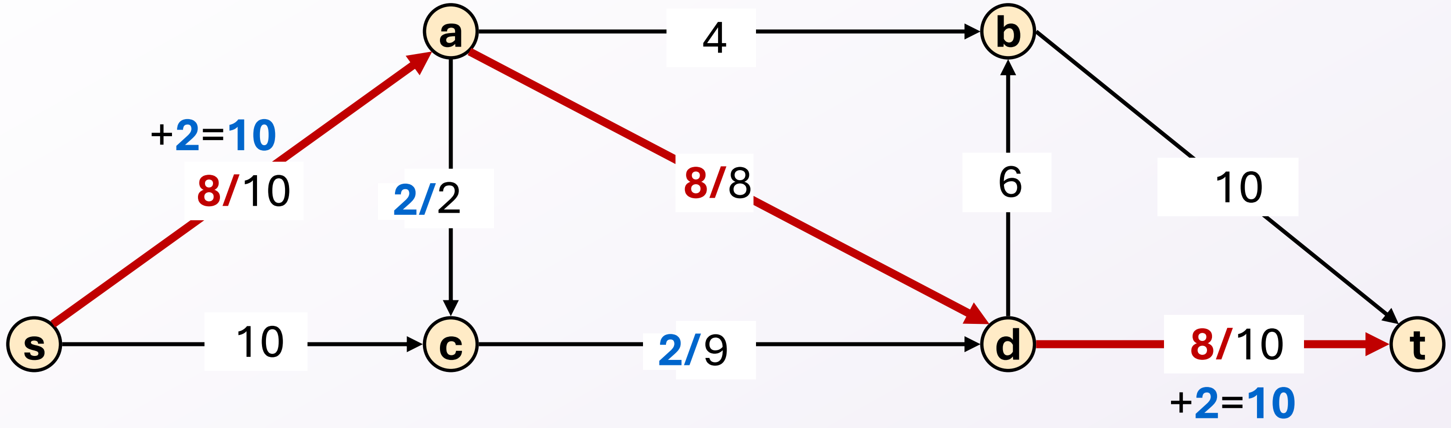

Flow value = **8**

$G_f$:



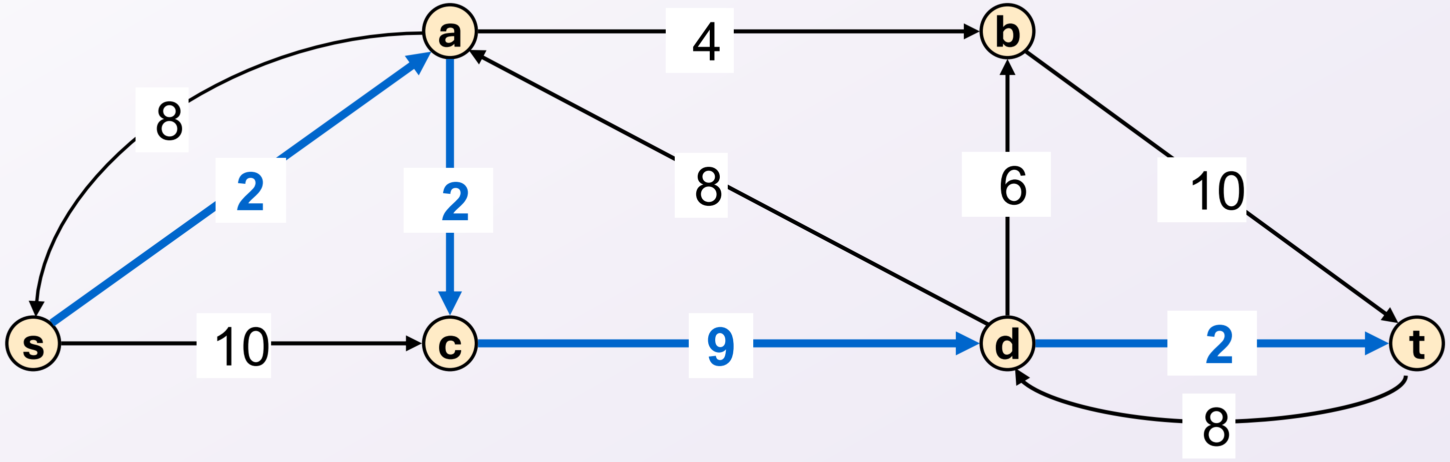
# Ford-Fulkerson Algorithm

$G$:



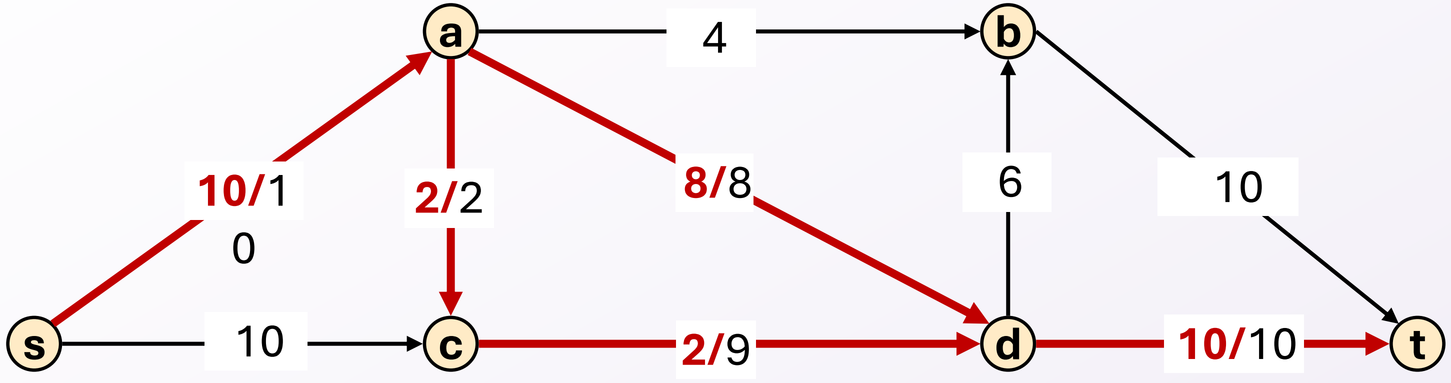

Flow value = 8
+2=10

$G_f$:



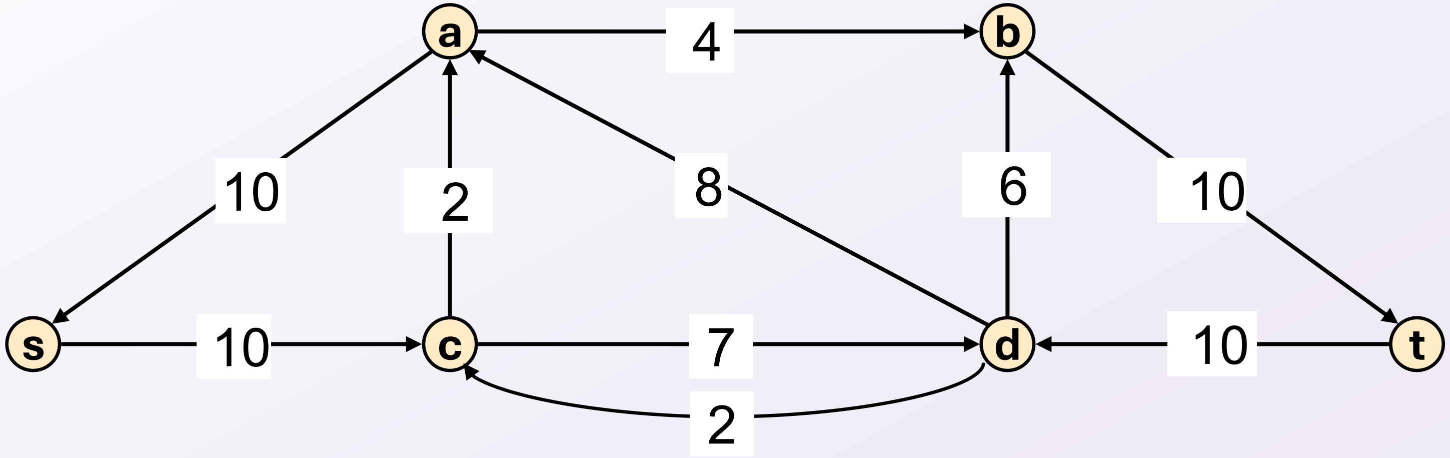
# Ford-Fulkerson Algorithm

$G$:



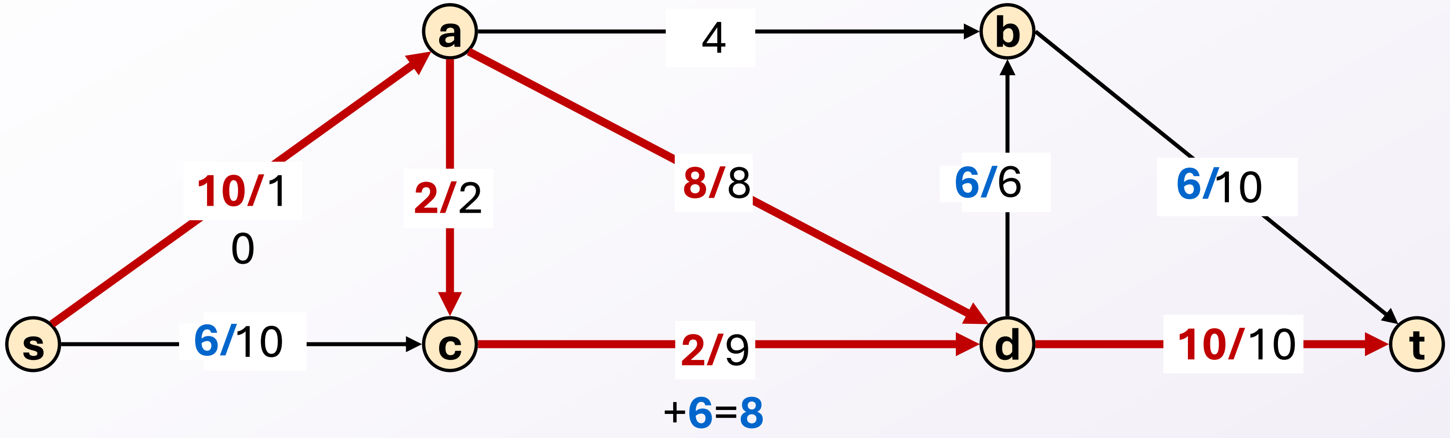

Flow value = **10**

$G_f$:



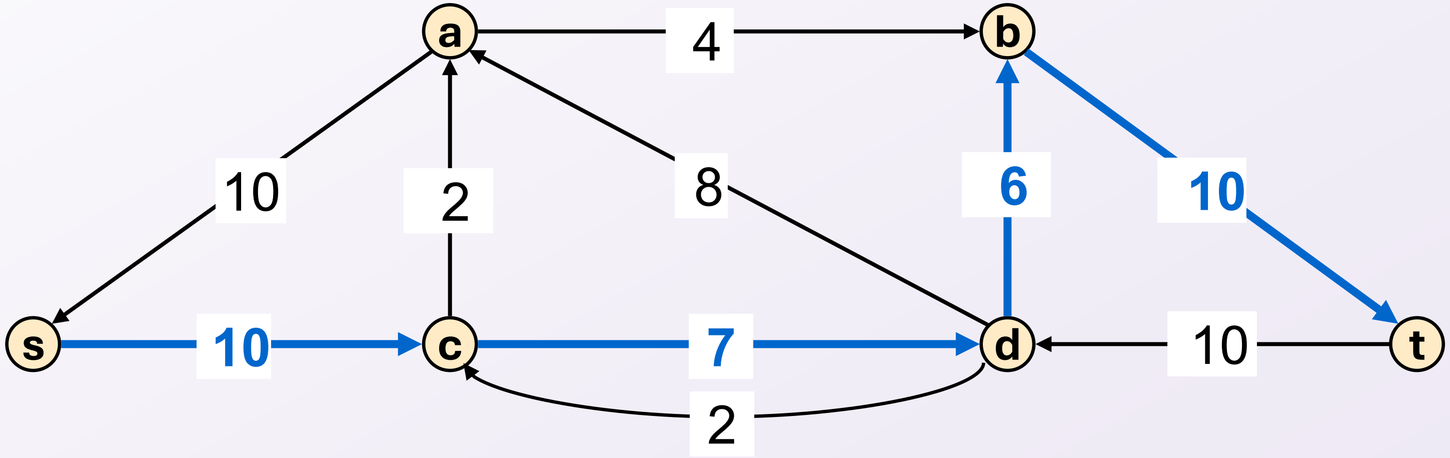
# Ford-Fulkerson Algorithm

$G$:



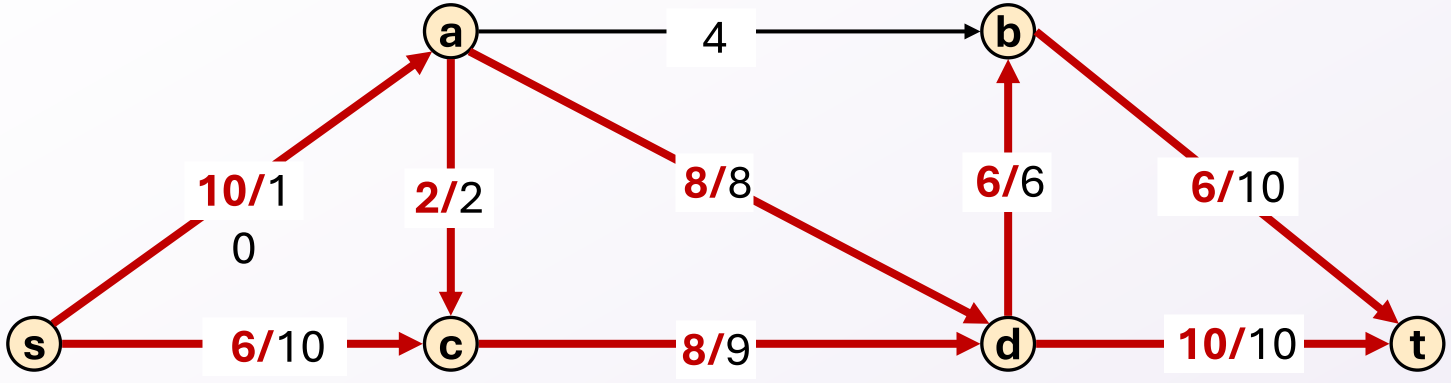

Flow value =
10
+6=16

$G_f$:



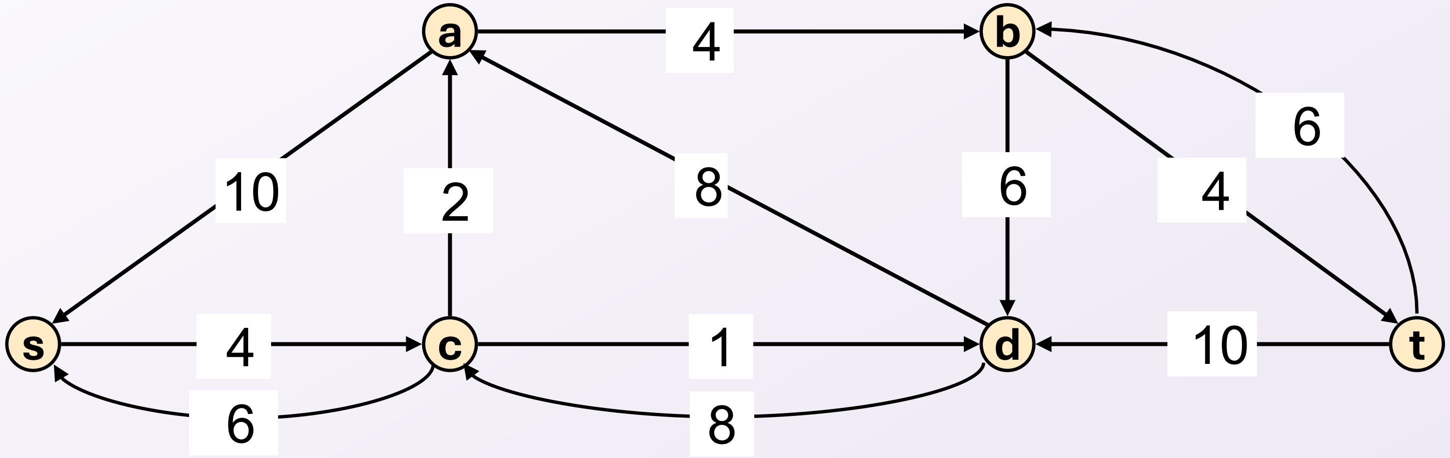
# Ford-Fulkerson Algorithm

$G$:



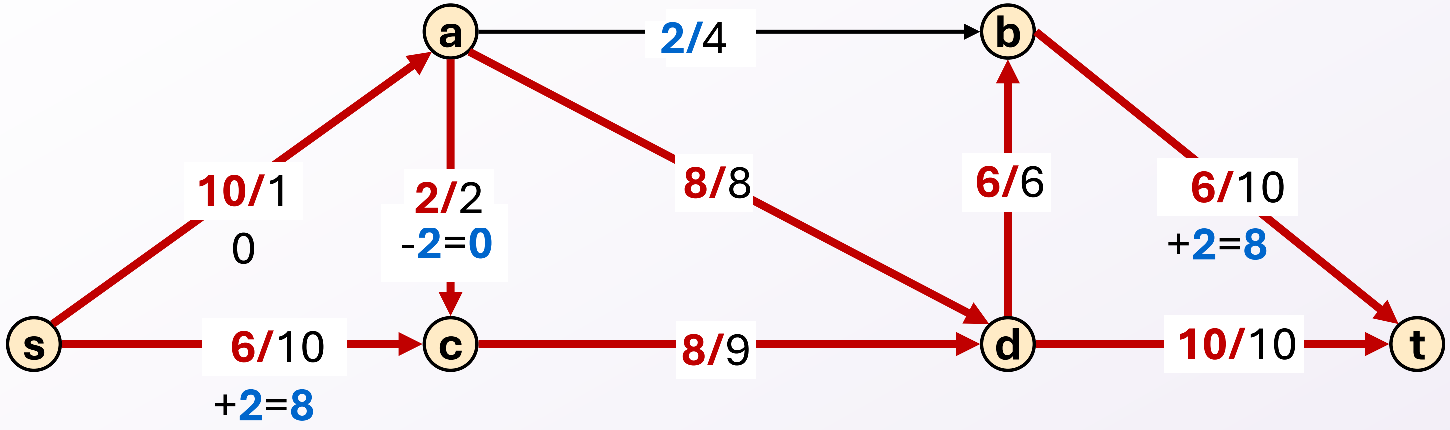

Flow value = **16**

$G_f$:



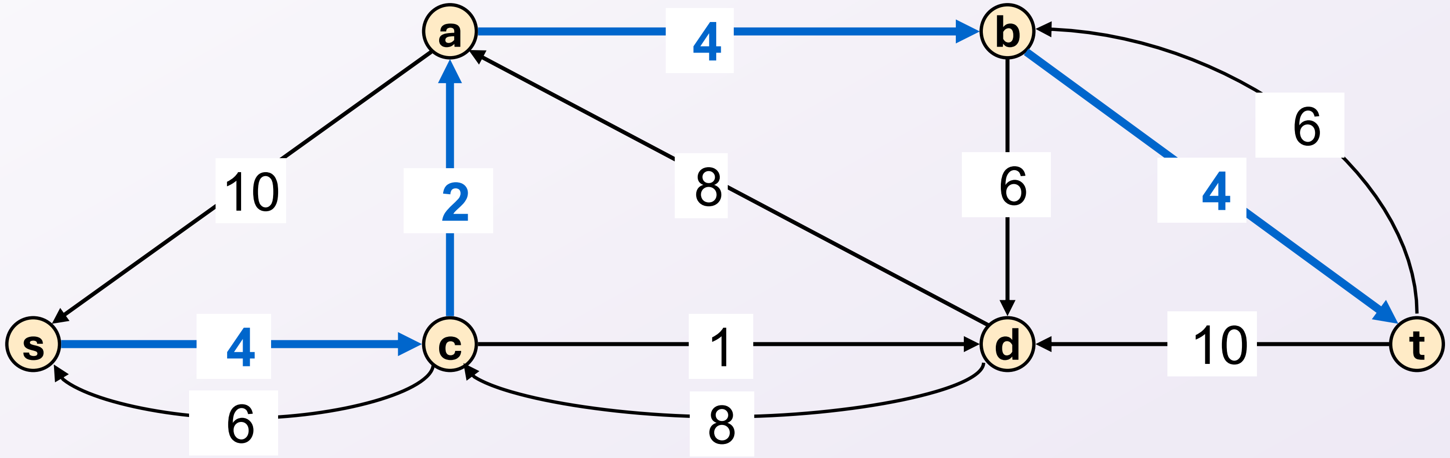
# Ford-Fulkerson Algorithm

$G$:



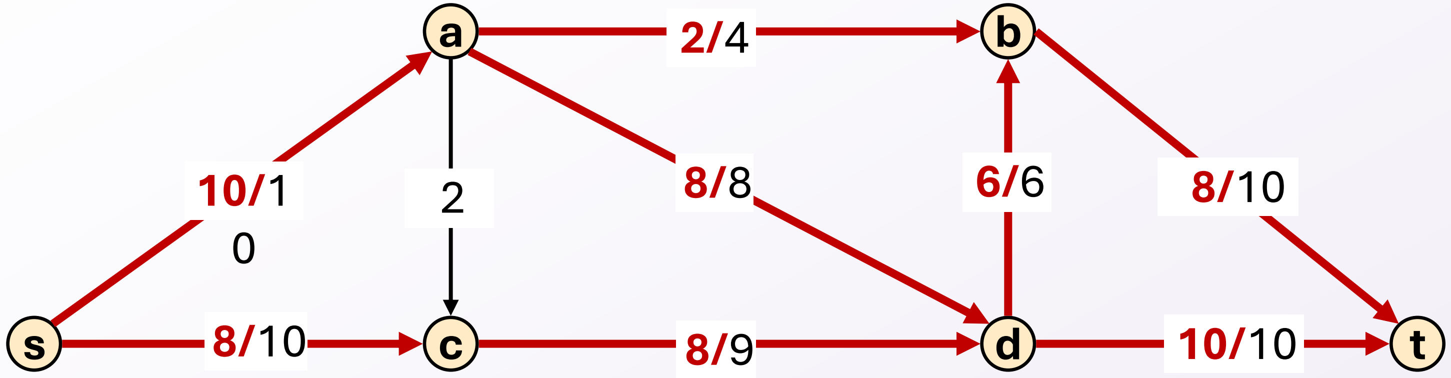

Flow value =
**16**
+**2**=**18**

$G_f$:



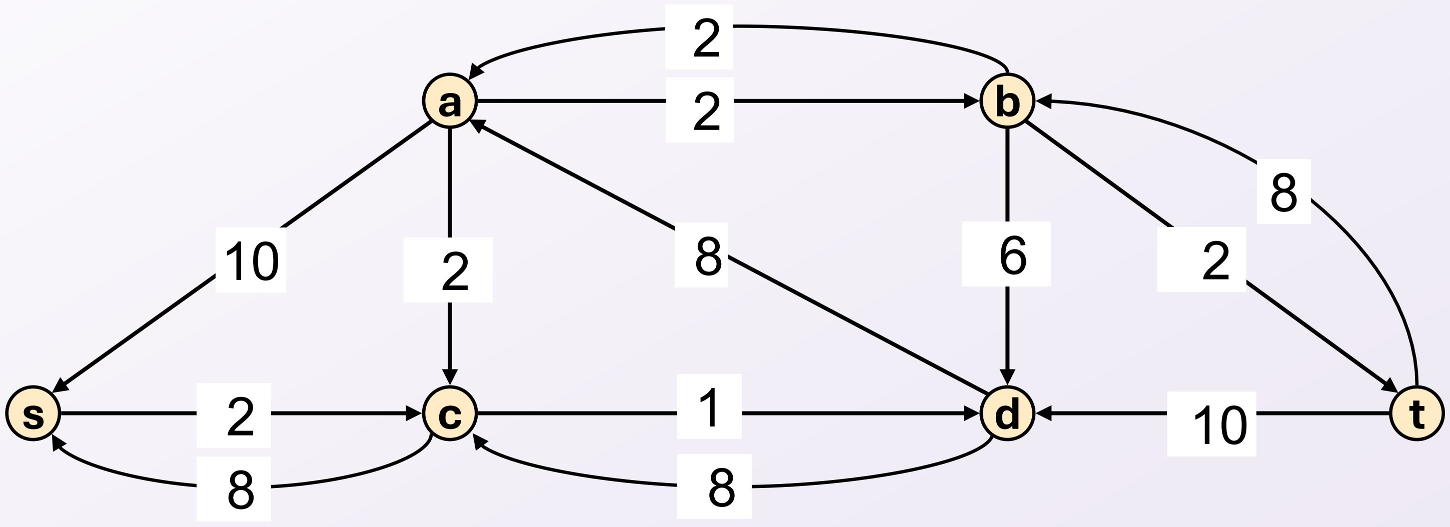
# Ford-Fulkerson Algorithm

$G$:



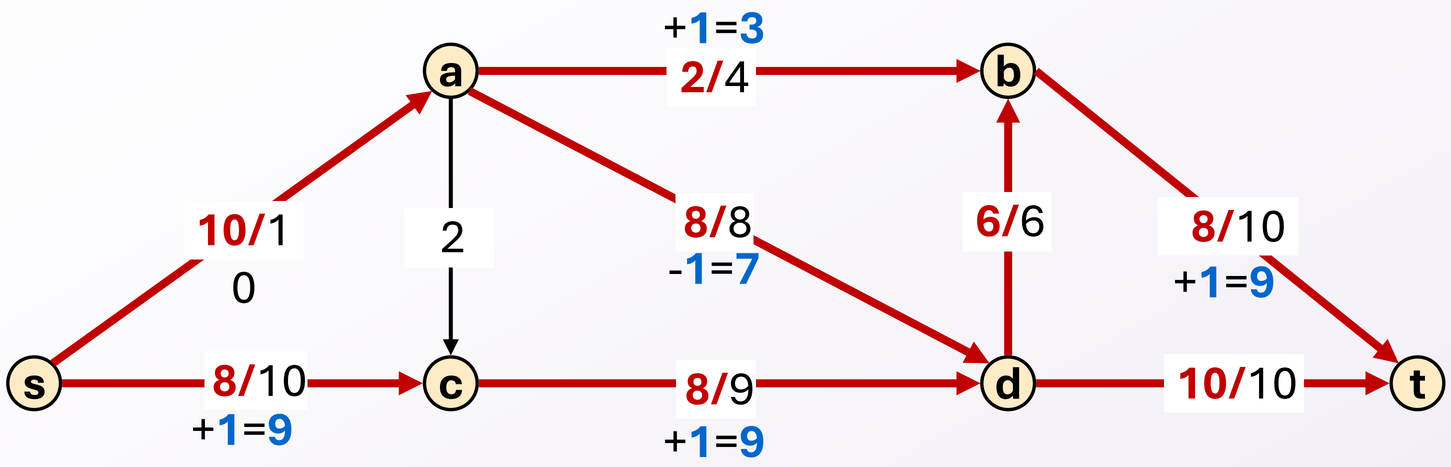

Flow value = **18**

$G_f$:



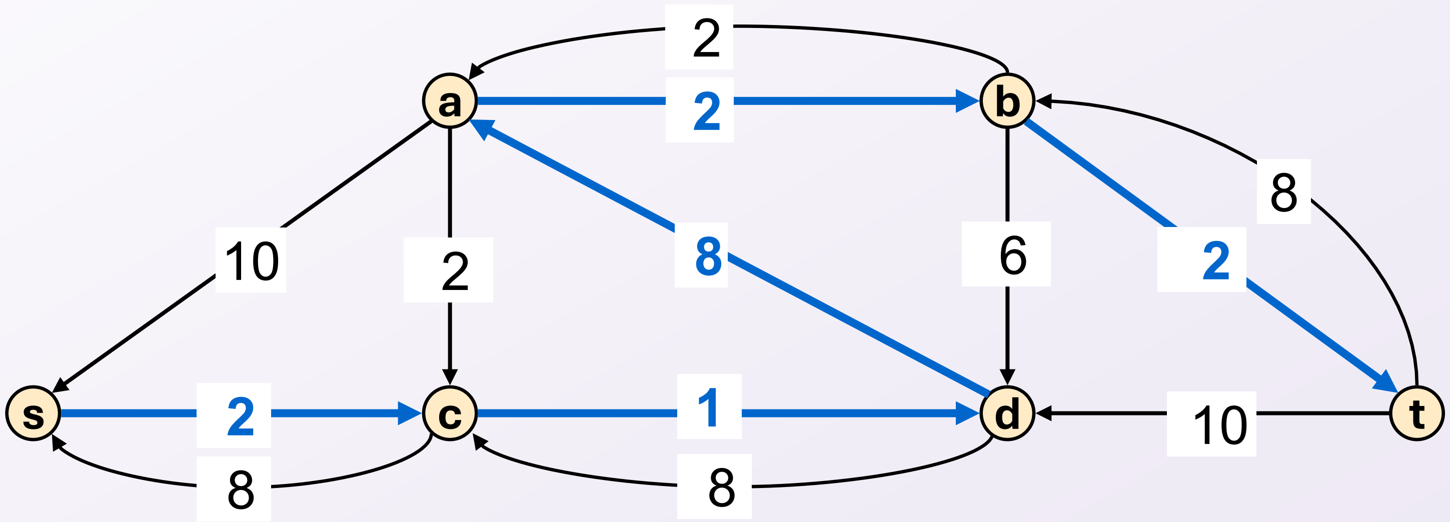
# Ford-Fulkerson Algorithm

$G$:



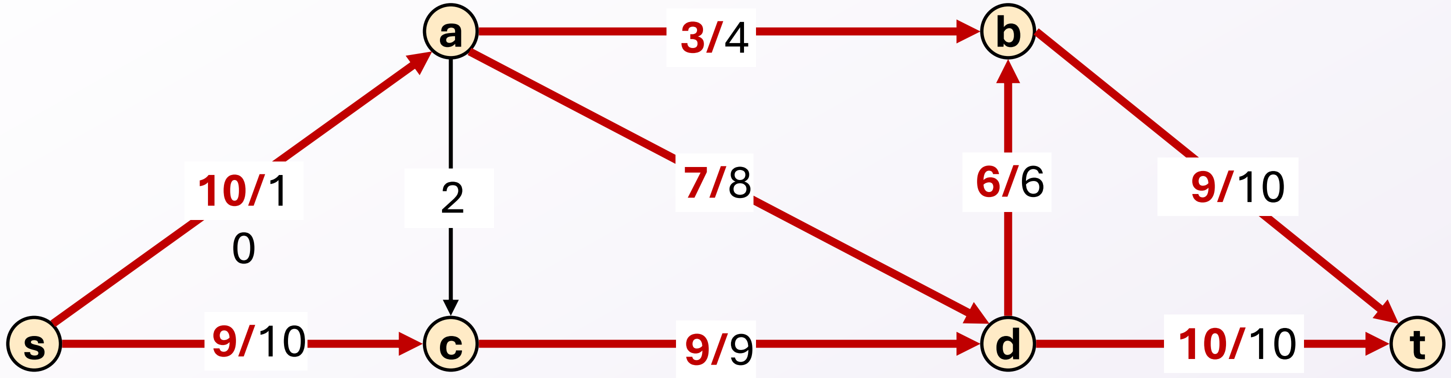

Flow value =
18
+1=19

$G_f$:



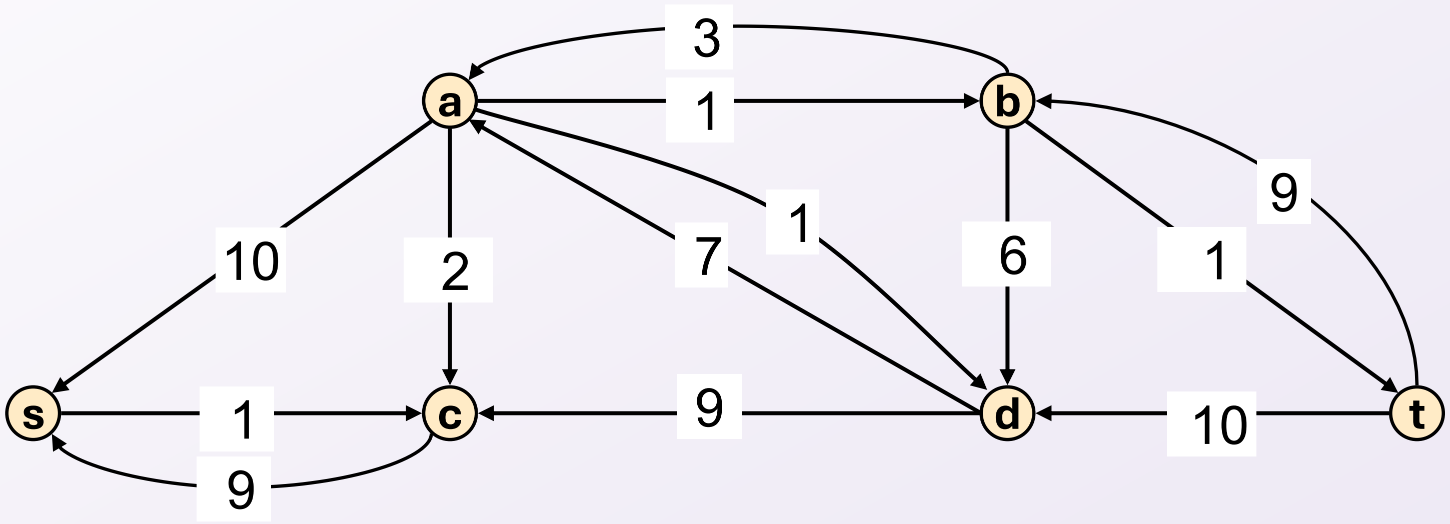
# Ford-Fulkerson Algorithm

$G$:




Flow value = **19**

$G_f$:



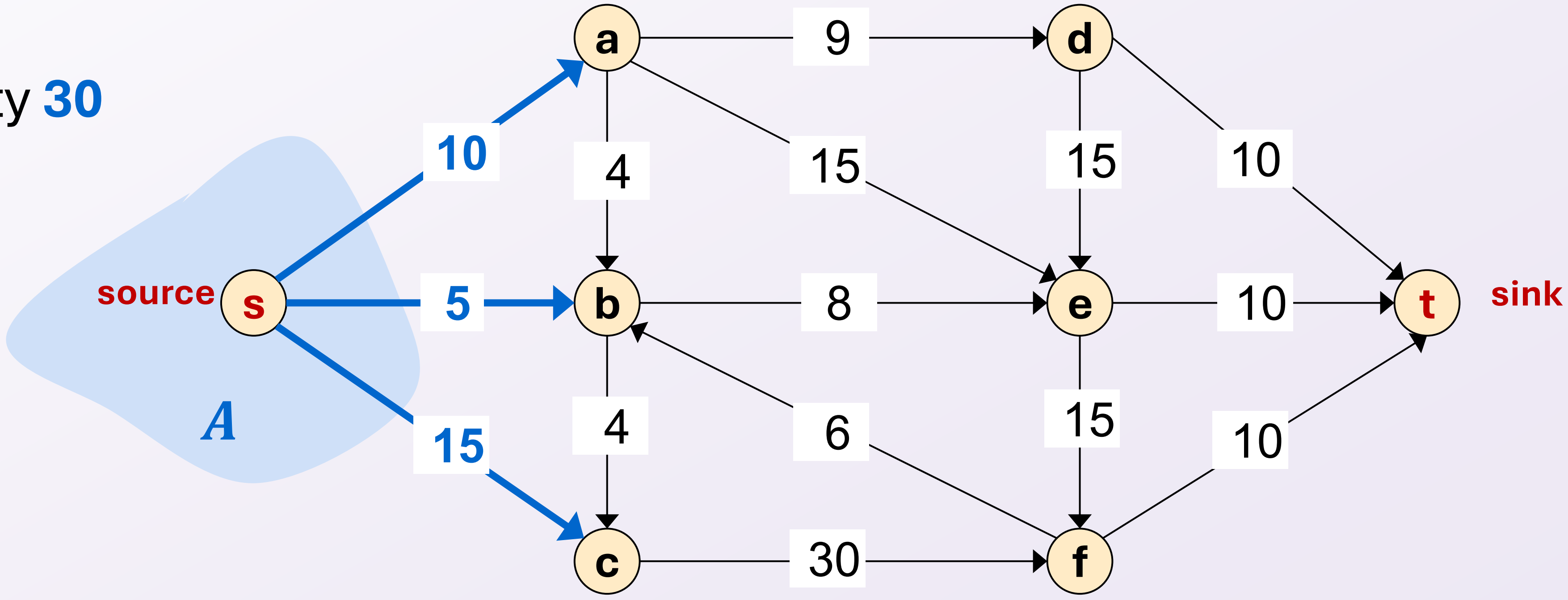
# Cuts

**Defn:** An $s$-$t$ cut is a partition $(A, B)$ of $V$ with $s \in A$ and $t \in B$.

The capacity of cut $(A, B)$ is

$$c(A, B) = \sum_{e \text{ out of } A} c(e)$$

capacity **30**



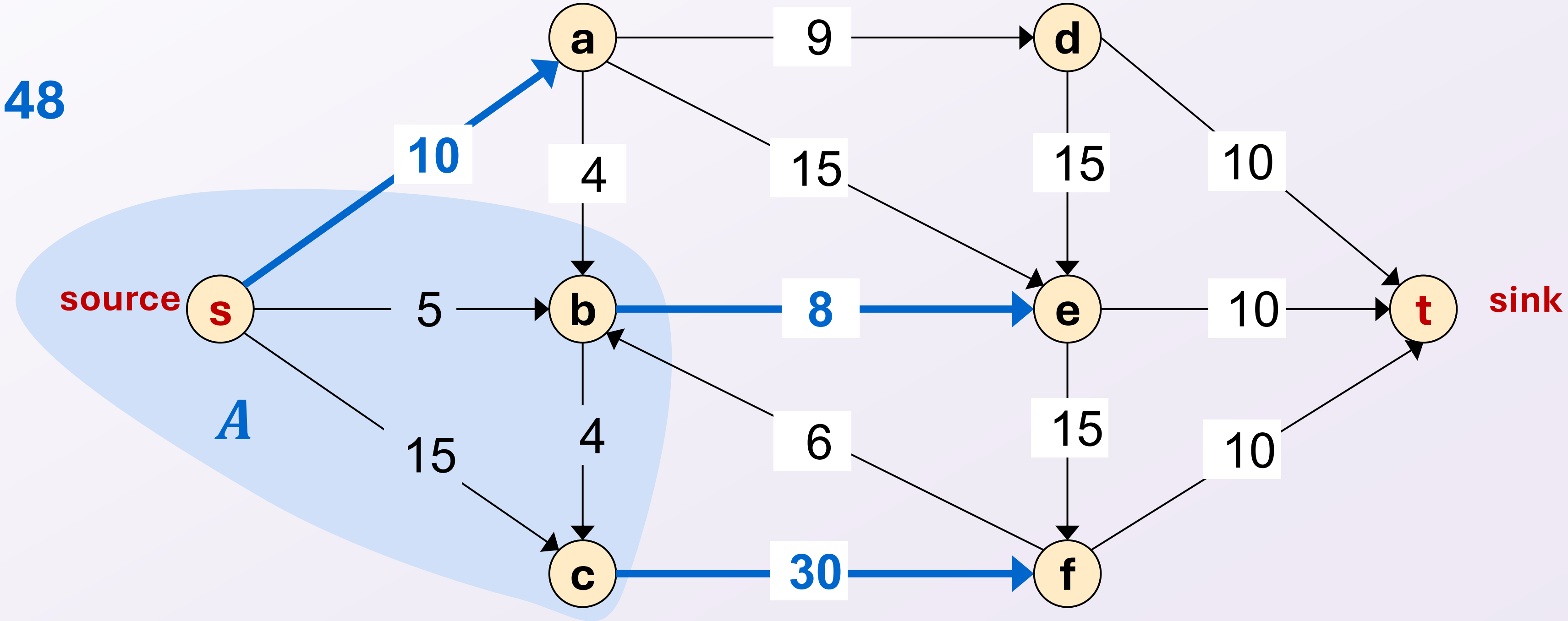
# Cuts

**Defn:** An $s$-$t$ cut is a partition $(A, B)$ of $V$ with $s \in A$ and $t \in B$.

The capacity of cut $(A, B)$ is

$$c(A, B) = \sum_{e \text{ out of } A} c(e)$$

capacity **48**



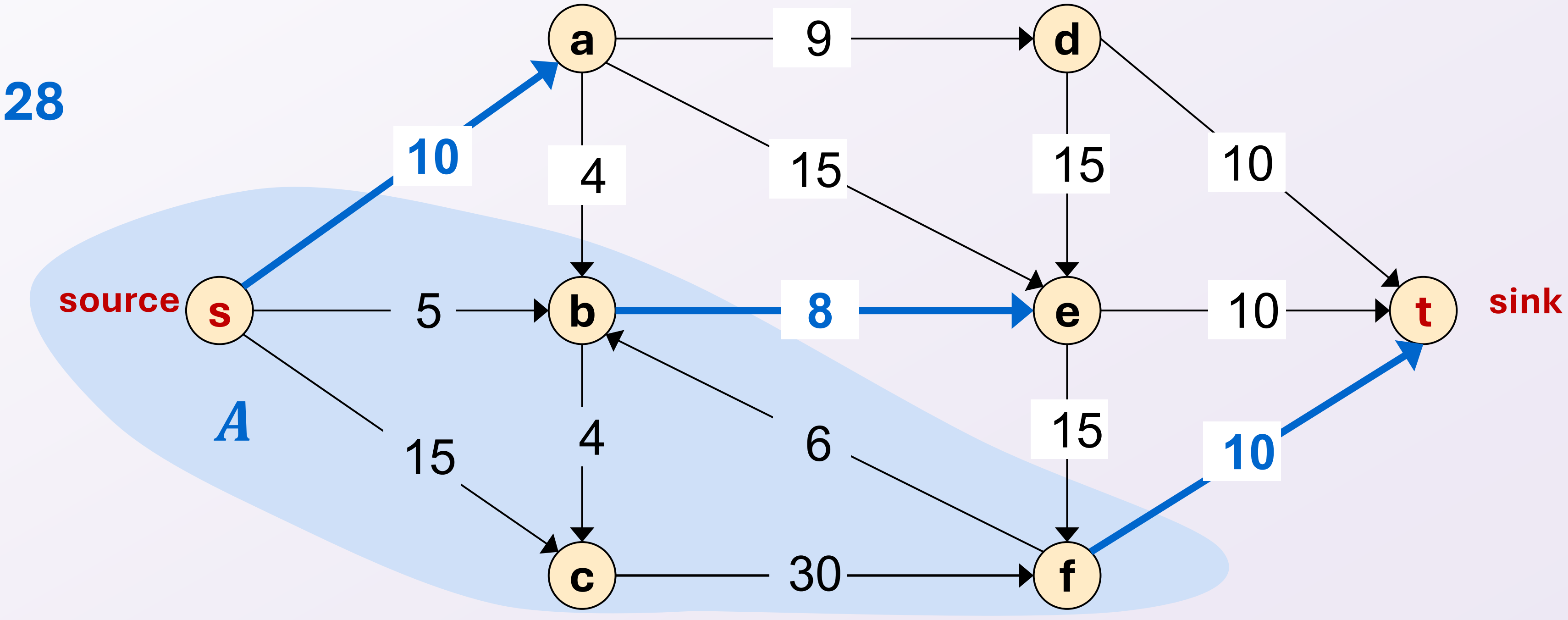
# Minimum Cut Problem

## Minimum s-t cut problem:

**Given:** a flow network

**Find:** an $s$-$t$ cut of minimum capacity

capacity **28**



# Flows and Cuts

Let $\boldsymbol{f}$ be any $\boldsymbol{s}$-$\boldsymbol{t}$ flow and $(\boldsymbol{A}, \boldsymbol{B})$ be any $\boldsymbol{s}$-$\boldsymbol{t}$ cut:

**Flow Value Lemma:** The net value of the flow sent across $(\boldsymbol{A}, \boldsymbol{B})$ equals $\boldsymbol{v}(\boldsymbol{f})$.

**Intuition**: All flow coming from $s$ must eventually reach $t$, and so must cross that cut

**Weak Duality:** The value of the flow is at most the capacity of the cut; i.e., $\boldsymbol{v}(\boldsymbol{f}) \leq \boldsymbol{c}(\boldsymbol{A}, \boldsymbol{B})$.

**Intuition**: Since all flow must cross any cut, any cut's capacity is an upper bound on the flow

**Corollary:** If $\boldsymbol{v}(\boldsymbol{f}) = \boldsymbol{c}(\boldsymbol{A}, \boldsymbol{B})$ then $\boldsymbol{f}$ is a maximum flow and $(\boldsymbol{A}, \boldsymbol{B})$ is a minimum cut.

**Intuition**: If we find a cut whose capacity matches the flow, we can't push more flow through that cut because it's already at capacity. We additionally can't find a smaller cut, since that flow was achievable.

# Flows and Cuts (Simplified)

The net flow crossing any cut equals the flow value.

Everything must cross the cut eventually

The capacity of any cut therefore is an upper bound on the max flow

No flow can exceed that capacity due to our previous observation

If we found a flow whose value matches the capacity of some cut, then we know that the flow must be maximum, and the cut must be minimum

If there was a smaller cut or larger flow, we've broken the previous point

# Flows and Cuts (Simplified)

1. The net flow crossing any cut equals the flow value.
    - Why? Everything must cross the cut eventually
2. The capacity of any cut therefore is an upper bound on the max flow
    - Why? No flow can exceed that capacity due to statement 1
3. If we found a flow whose value matches the capacity of some cut, then we know that the flow must be maximum, and the cut must be minimum
    - Why? If there was a smaller cut or larger flow, we've broken statement 2

**What we need for correctness:**

**When Ford-Fulkerson terminates, there exists a cut whose capacity matches the current flow value.**

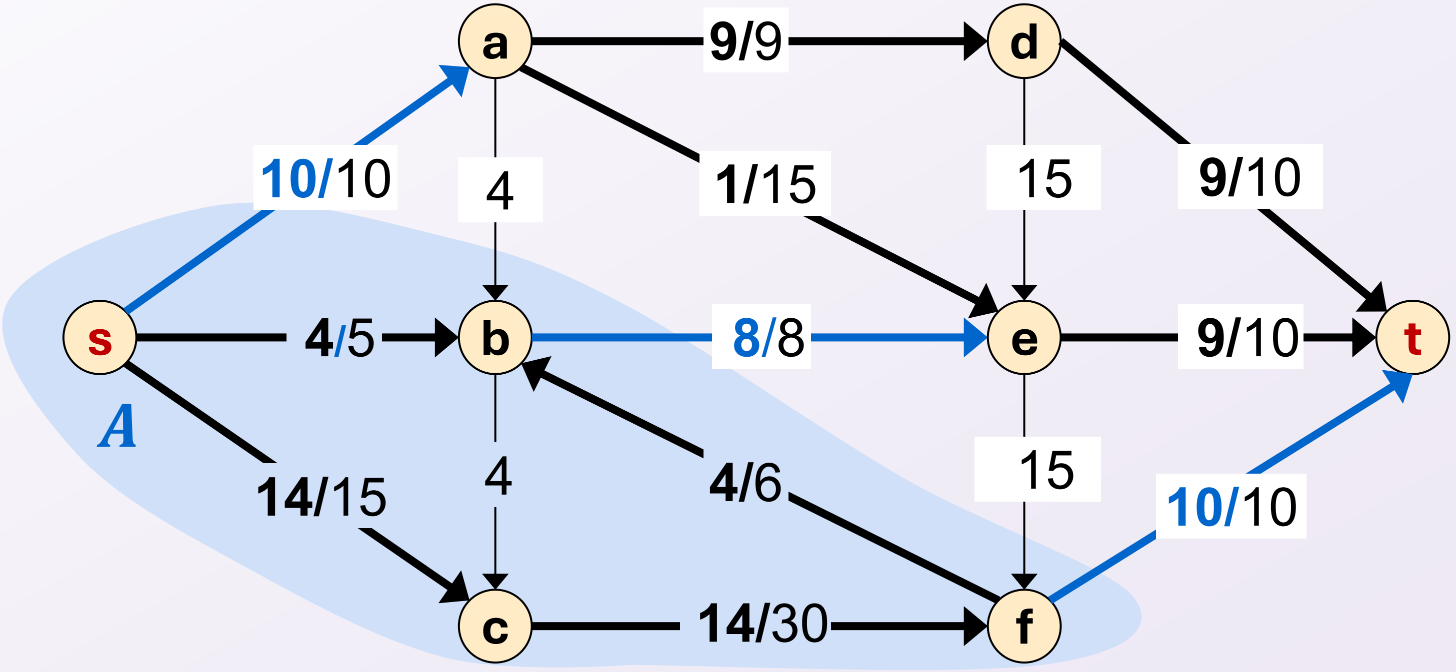
# Certificate of Optimality

Let $f$ be any $s$-$t$ flow and $(A, B)$ be any $s$-$t$ cut.

If $v(f) = c(A, B)$ then $f$ is a max flow and $(A, B)$ is a min cut.

Value of flow = **28**

Capacity of cut = **28**




Both are optimal!
Each “certified” correctness of the other!



32

# Identifying the cut

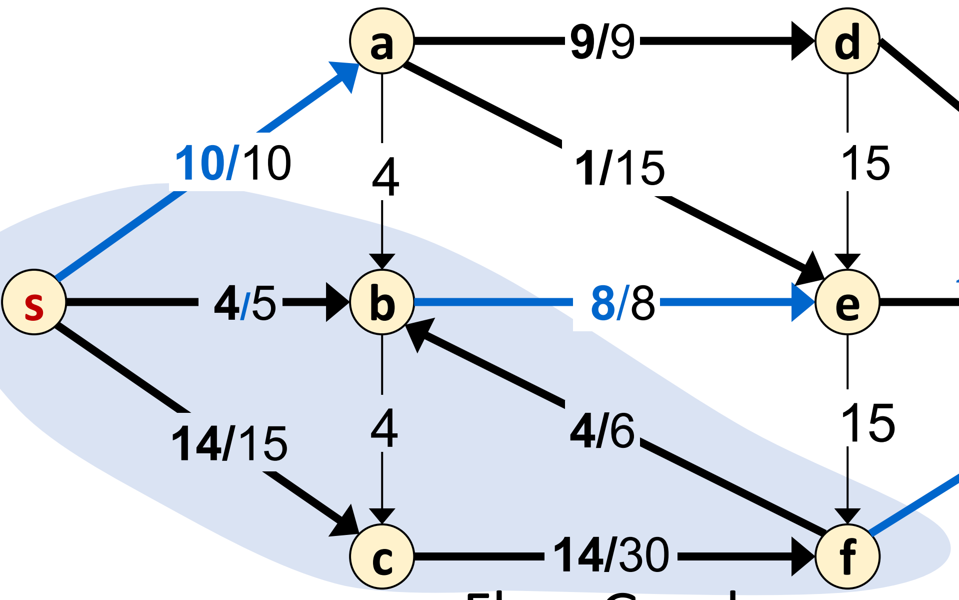
**To Show:** If there is no augmenting path w.r.t. $f$, there is a cut $(A, B)$ s.t. $v(f) = c(A, B)$.

**Selecting a cut:** Let $f$ be a flow with no augmenting paths.

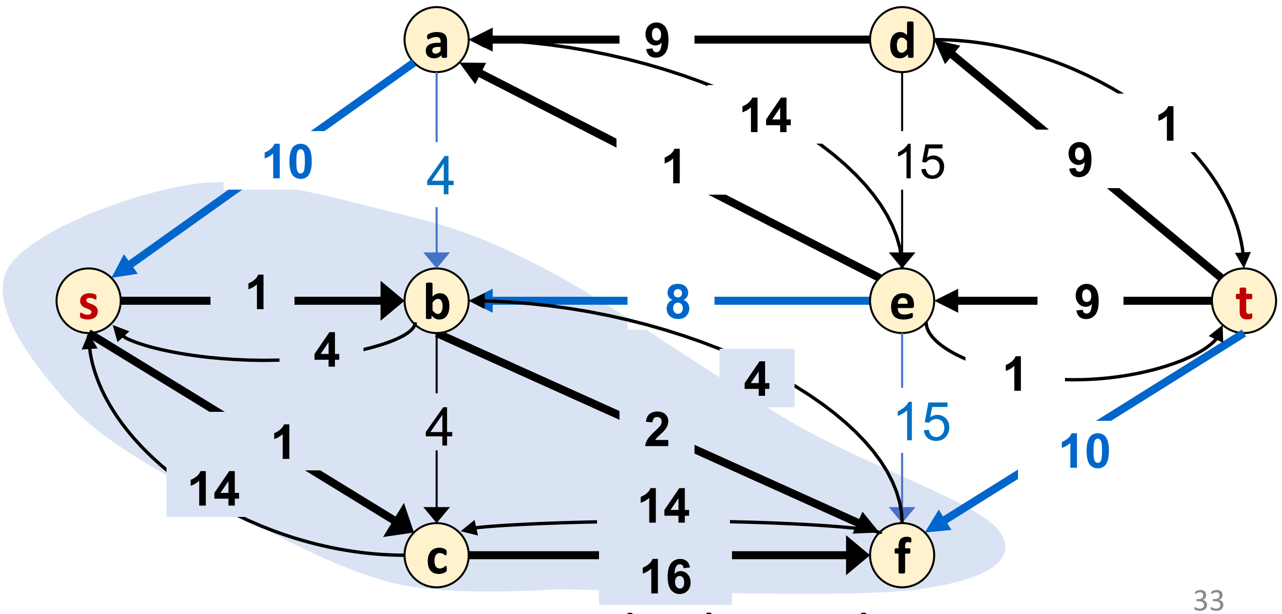
Let $A$ be the set of vertices reachable from $s$ in residual graph $G_f$.

- By definition of $A$, $s \in A$.
- Since no augmenting path ($s$-$t$ path in $G_f$), $t \notin A$.

Notice:
- all edges out of the cut are saturated (flow=capacity)
- all edges into the cut have no flow




Flow Graph




Residual Graph

# Flow Value = Cut Capacity

**To Show:** If there is no augmenting path w.r.t. $f$, there is a cut $(A, B)$ s.t. $v(f) = c(A, B)$.

**The cut:** $A$ is the set of all nodes reachable from $s$ in the residual graph

$B$ is the set of all the other nodes in the graph

**Showing Flow value = Cut Capacity:**

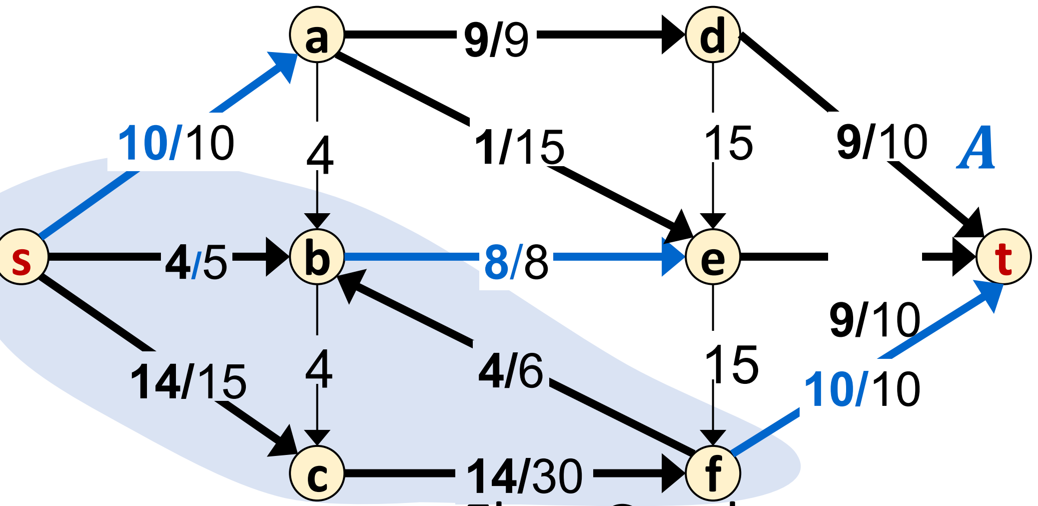
For any edge outgoing from $A$ to $B$, that edge is saturated (flow = capacity)

Otherwise there would be an edge in the residual graph for the remaining capacity. Contradiction!

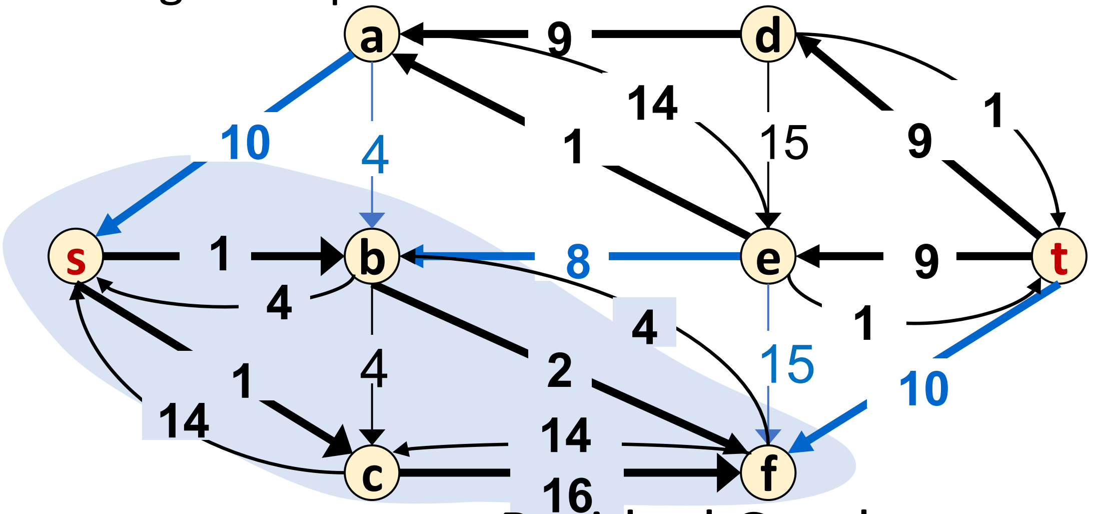
For any edge incoming from $B$ to $A$, that edge has no flow (flow = 0)

Otherwise there would be an edge in the residual graph to undo the flow. Contradiction!

So summing the flows of all $A$ to $B$ edges is the same as summing the capacities!




Flow Graph




Residual Graph

# Flows and Cuts (Complete)

1. The net flow crossing any cut equals the flow value.
    - Why? Everything must cross the cut eventually
2. The capacity of any cut therefore is an upper bound on the max flow
    - Why? No flow can exceed that capacity due to statement 1
3. If we found a flow whose value matches the capacity of some cut, then we know that the flow must be maximum, and the cut must be minimum
    - Why? If there was a smaller cut or larger flow, we've broken statement 2
4. When Ford Fulkerson terminates, there is a cut whose capacity matches the flow
    - Why? Select one side of the cut to be nodes reaching from $s$ in the residual graph, the other side to be the rest of the nodes. That cut's capacity matches the flow value.
    - Thus the cut is minimum, and the flow is maximum!

# Final reminders

HW6 released, due Friday @ 11:59pm.

Quiz 2 on Friday 11/21 in class

look for quiz prep materials on Friday

I have OH now-12:30pm:

- Meet at front of classroom, we'll walk over together
- CSE (Allen) 434 if you're coming later

Glenn has online OH 12–1pm