CSE 421 Winter 2025
Lecture 16:
Max Flow Min Cut

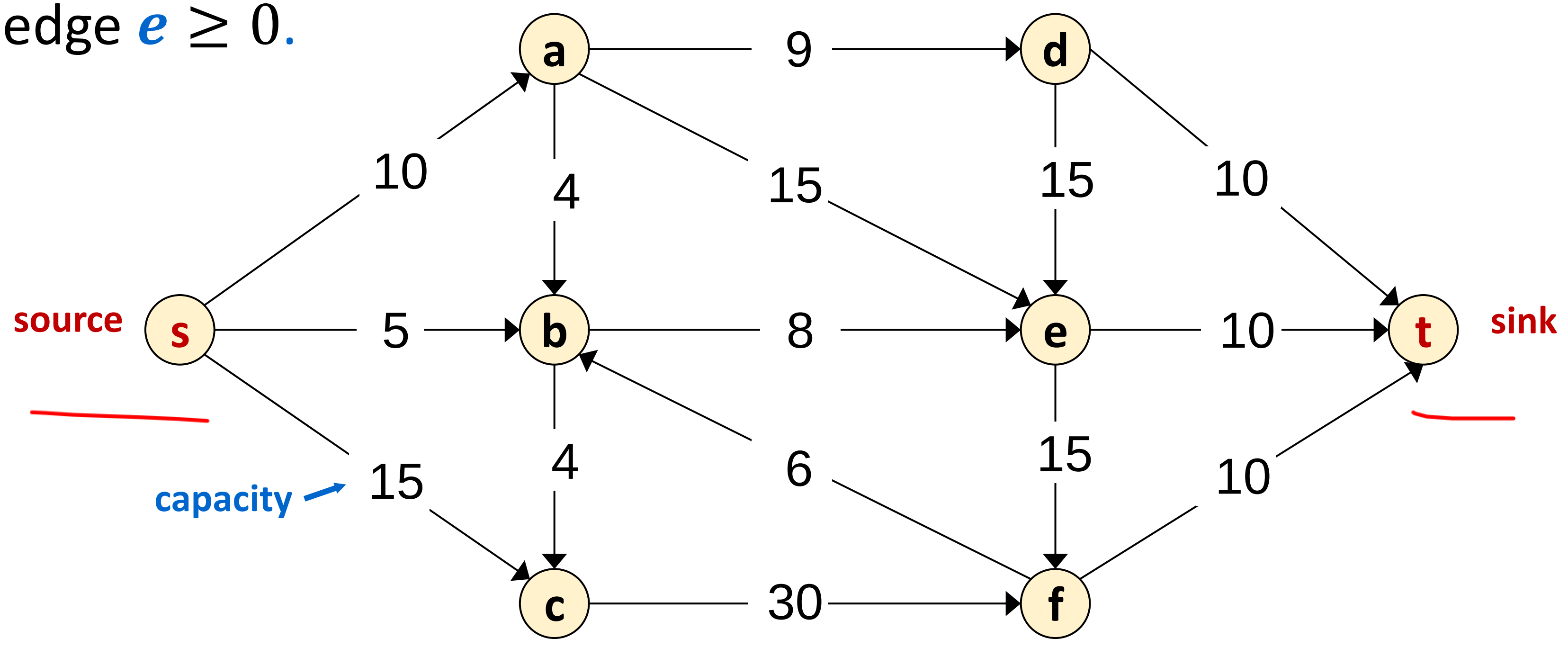
Nathan Brunelle

http://www.cs.uw.edu/421

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 $e \geq 0$.



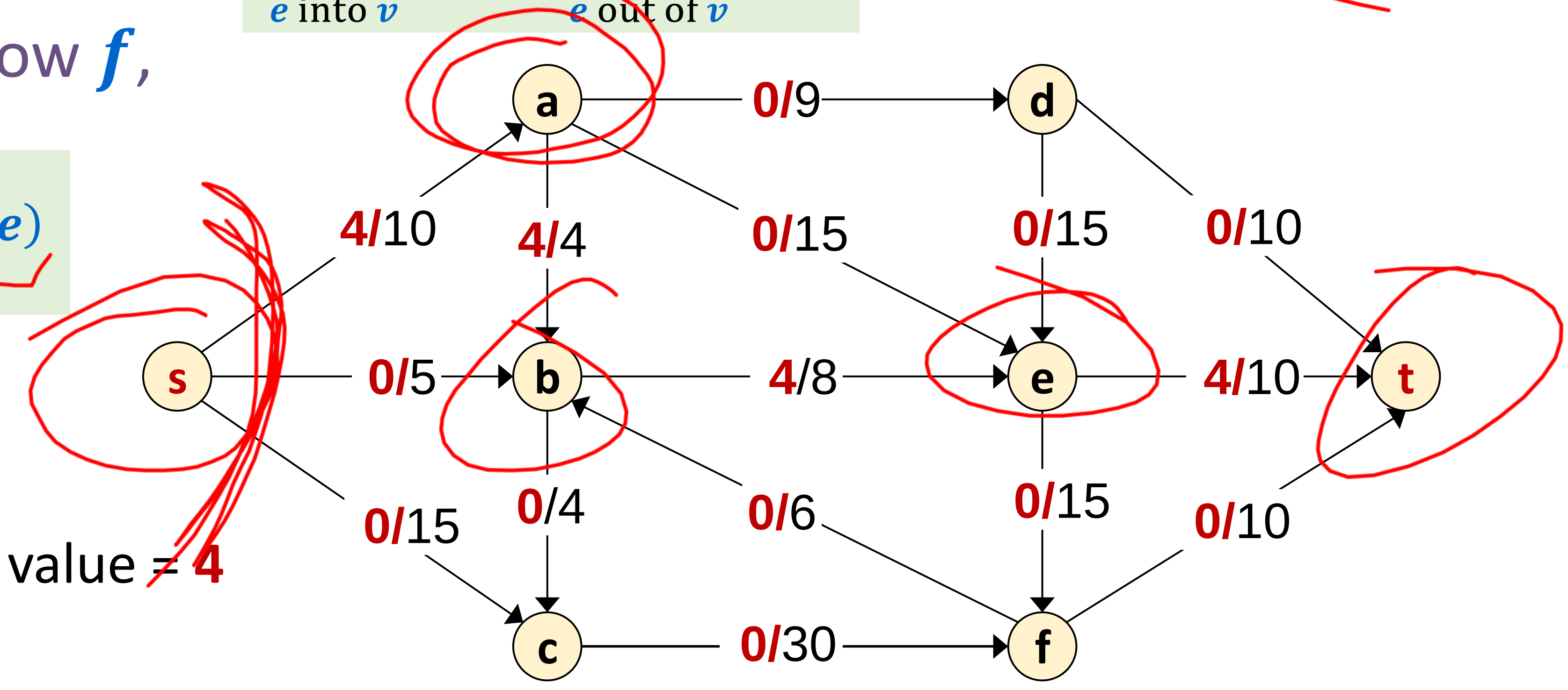
Flows

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 $v \in V - \{s, t\}$: $\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$ [flow conservation]

Defn: The value of flow f,

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



Flows

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

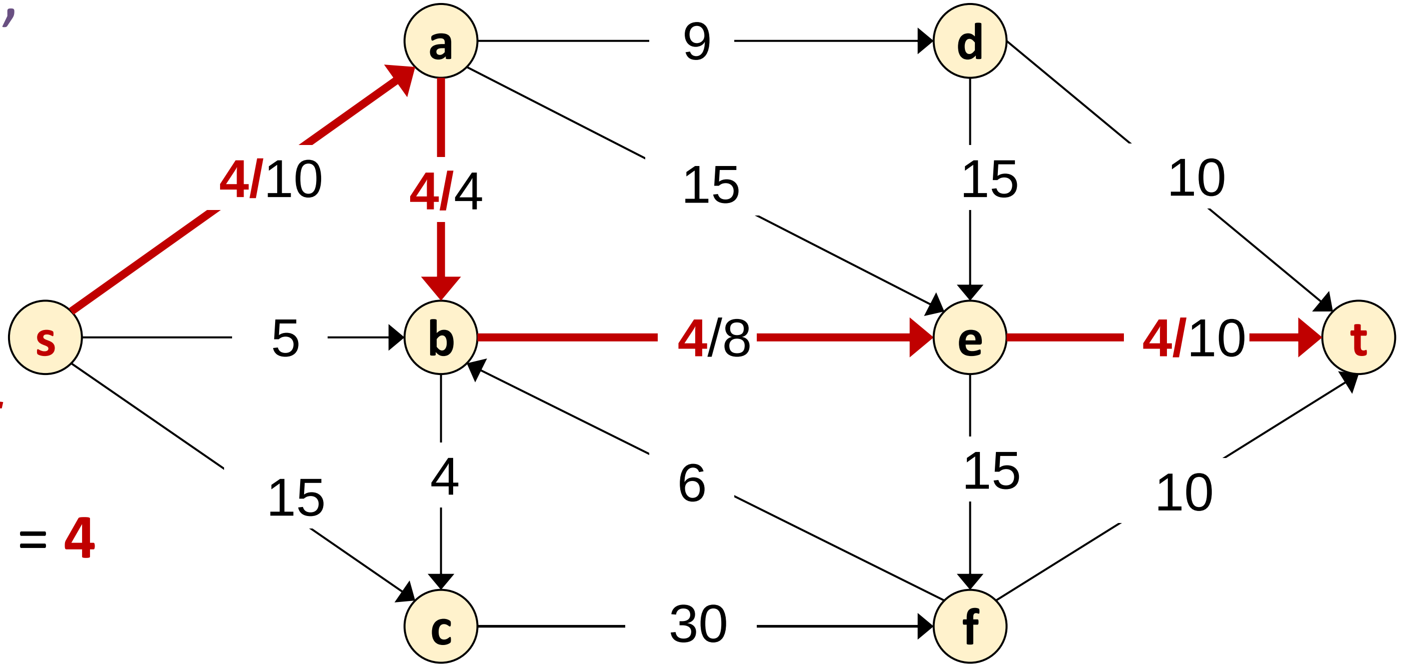
- For each $e \in E$: $0 \leq f(e) \leq c(e)$ [capacity constraints]
- For each $v \in V - \{s, t\}$: $\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$ [flow conservation]

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**



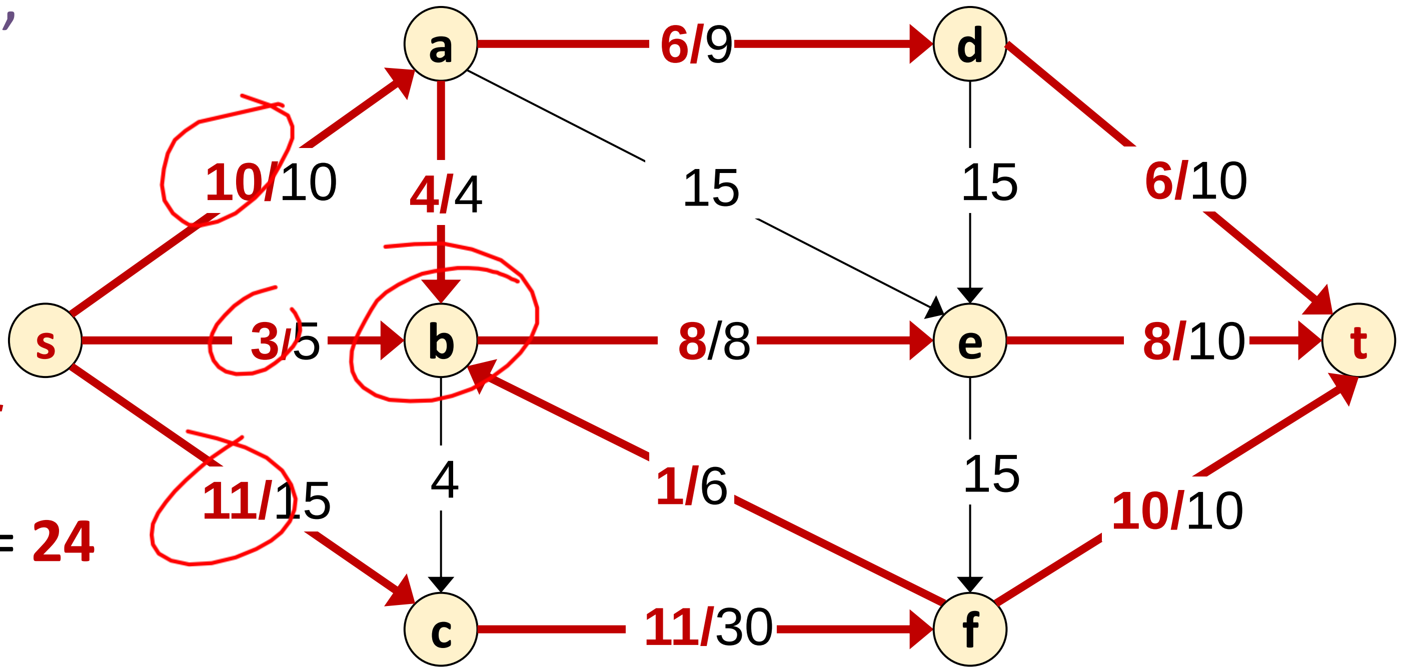
Flows

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 $v \in V - \{s, t\}$: $\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$ [flow conservation]

Defn: The value of flow f,

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



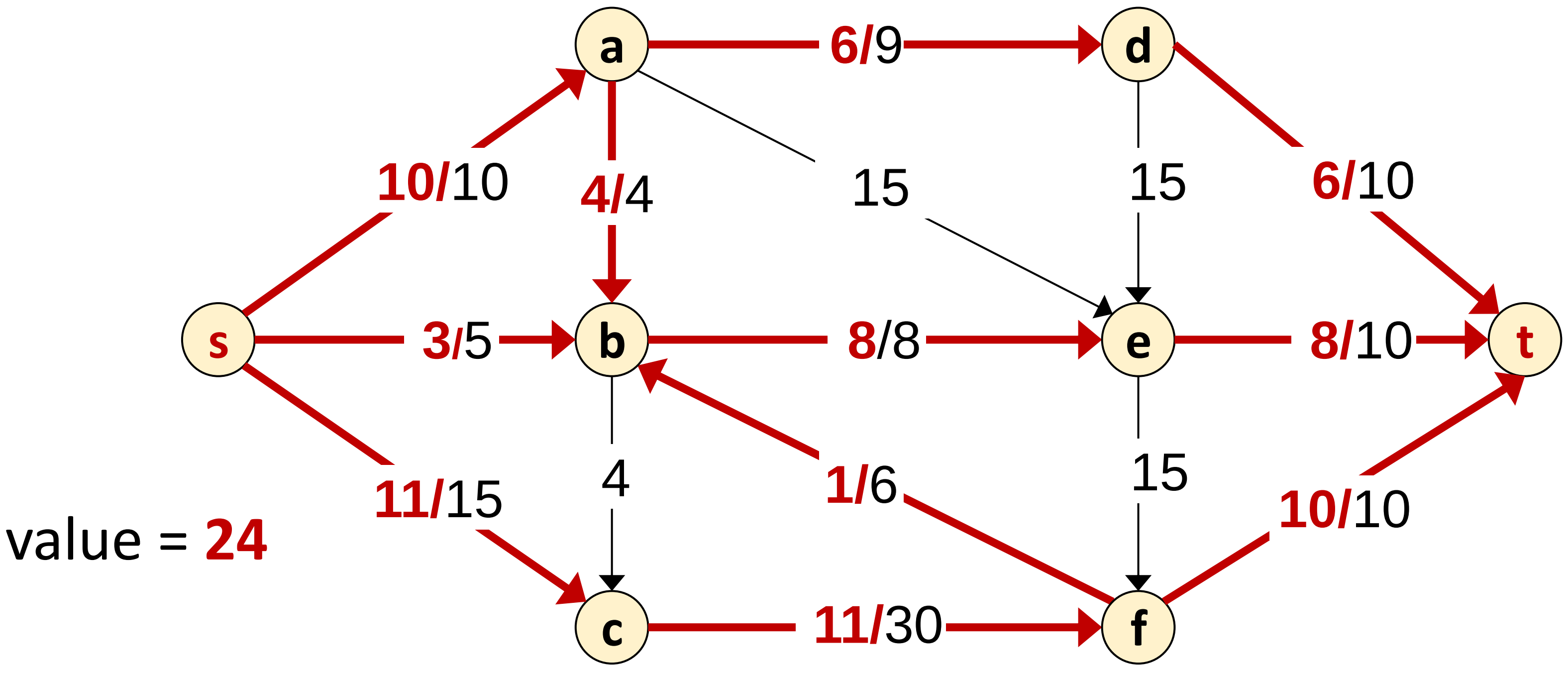
Only show non-zero values of f

value = **24**

Maximum Flow Problem

Given: a flow network

Find: an s-t flow of maximum value

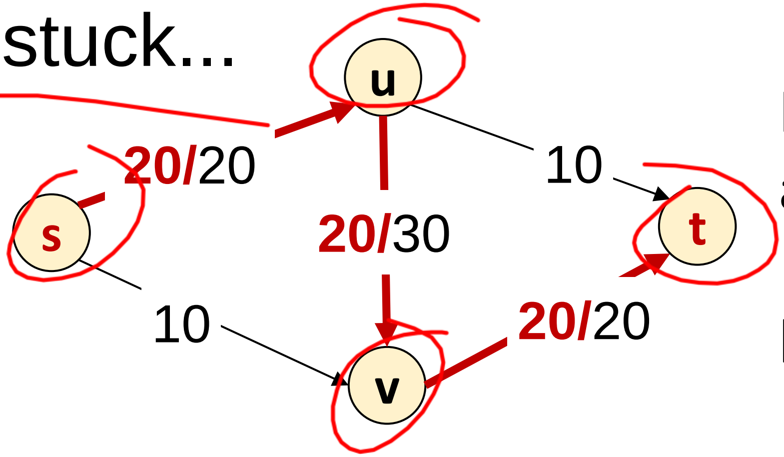


Towards a Max Flow Algorithm

What about the following greedy algorithm?

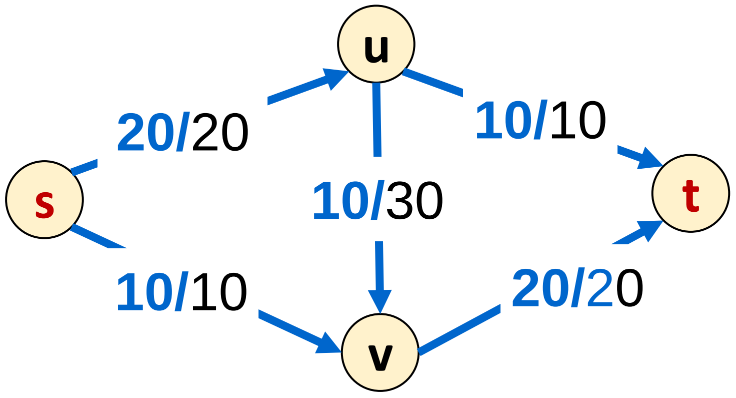
- Start with $\boldsymbol{f(e) = 0}$ for all edges $\boldsymbol{e \in E}$.
- While there is an $\boldsymbol{s}$-$\boldsymbol{t}$ path $\boldsymbol{P}$ where each edge has $\boldsymbol{f(e) < c(e)}$.
  - "Augment" flow along $\boldsymbol{P}$; that is:
    - Let $\boldsymbol{\alpha} = \min_{\boldsymbol{e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

Can get stuck...



Has flow value **20** and no path $\boldsymbol{P}$

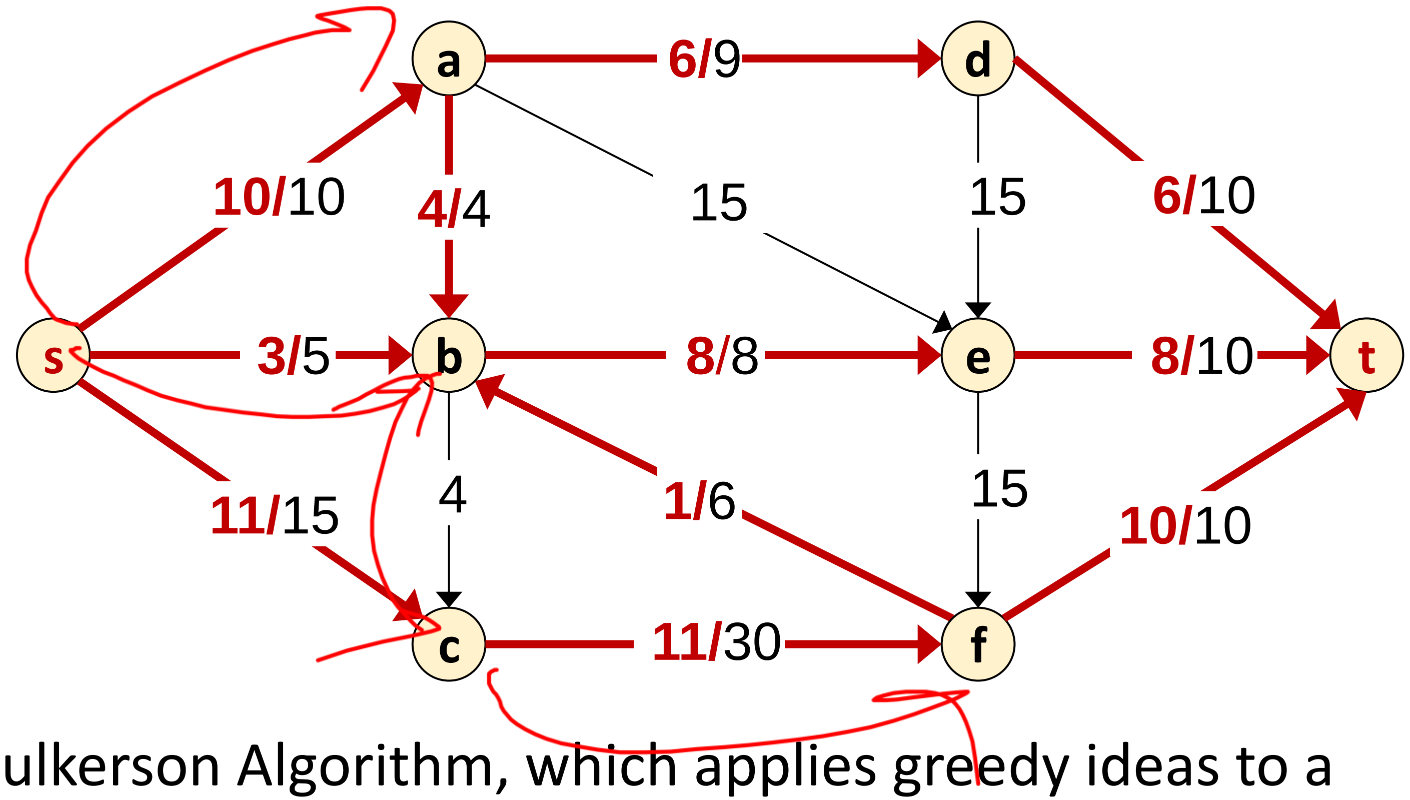
but **30** is possible



Another “Stuck” Example

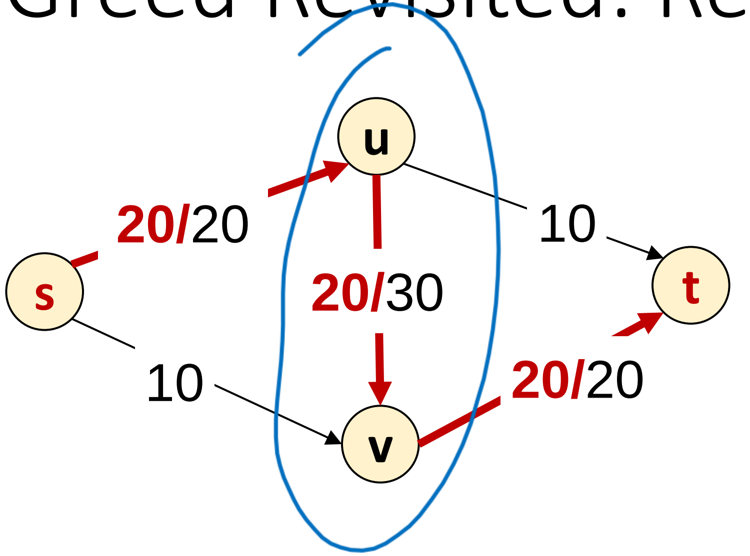
On every s-t path there is some edge with $f(e) = c(e)$:

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

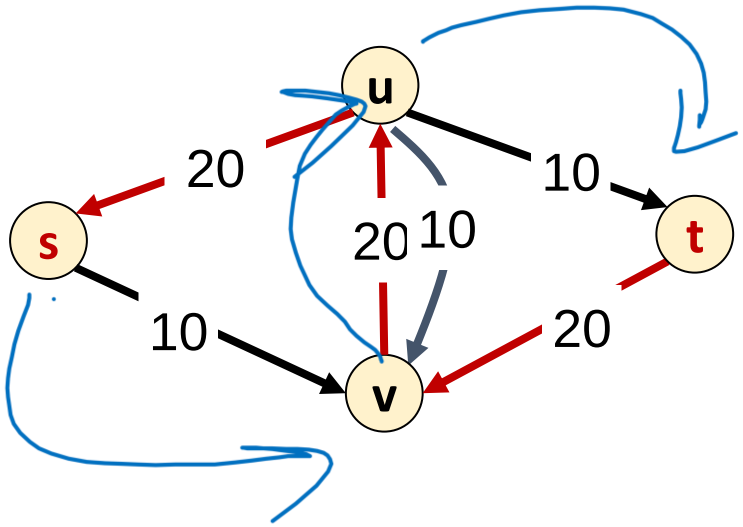


The only way we could route more flow from **s** to **t** would be to reduce the flow from **u** to **v** to make room for that amount of extra flow from **s** to **v**.
But to conserve flow we also would need to increase the flow from **u** to **t** by that same amount.

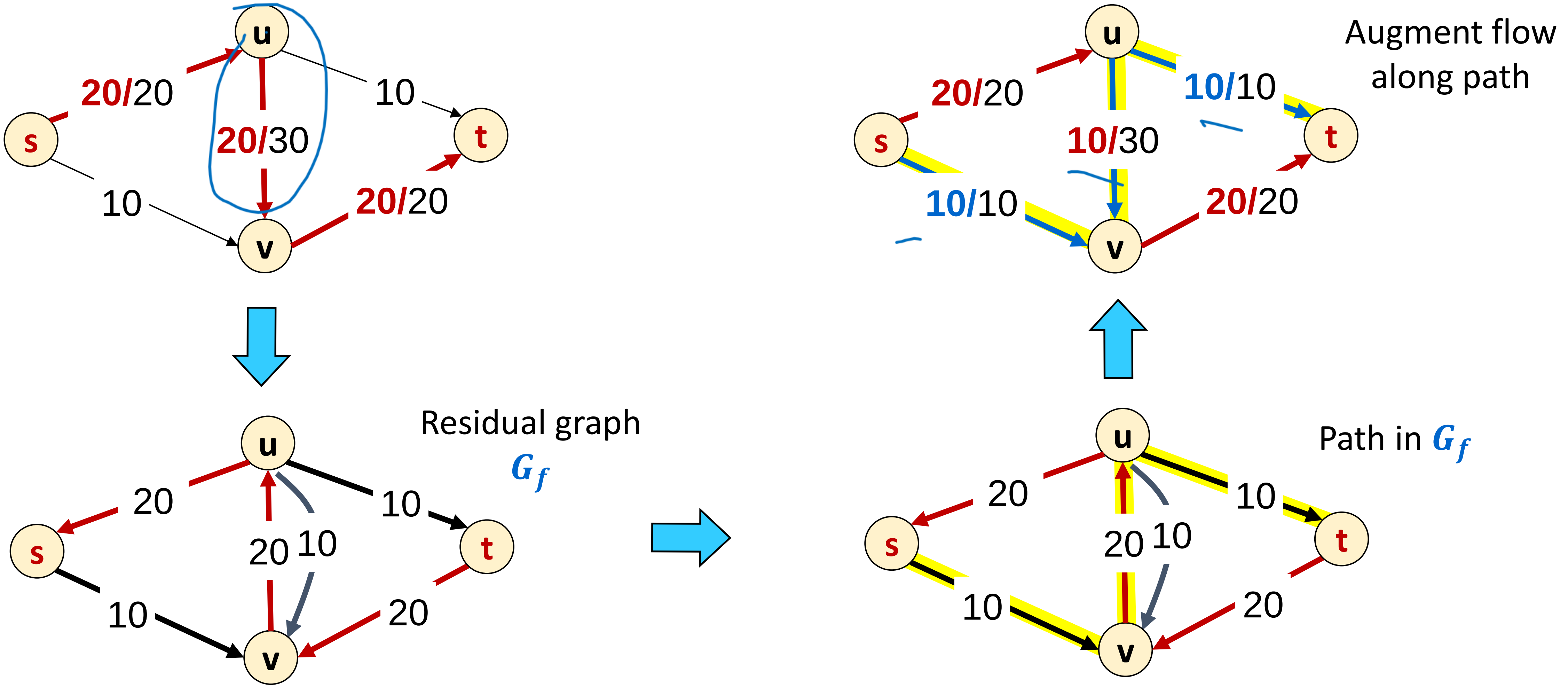
Suppose that we took this flow f as a baseline, what changes could each edge handle?

- 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**

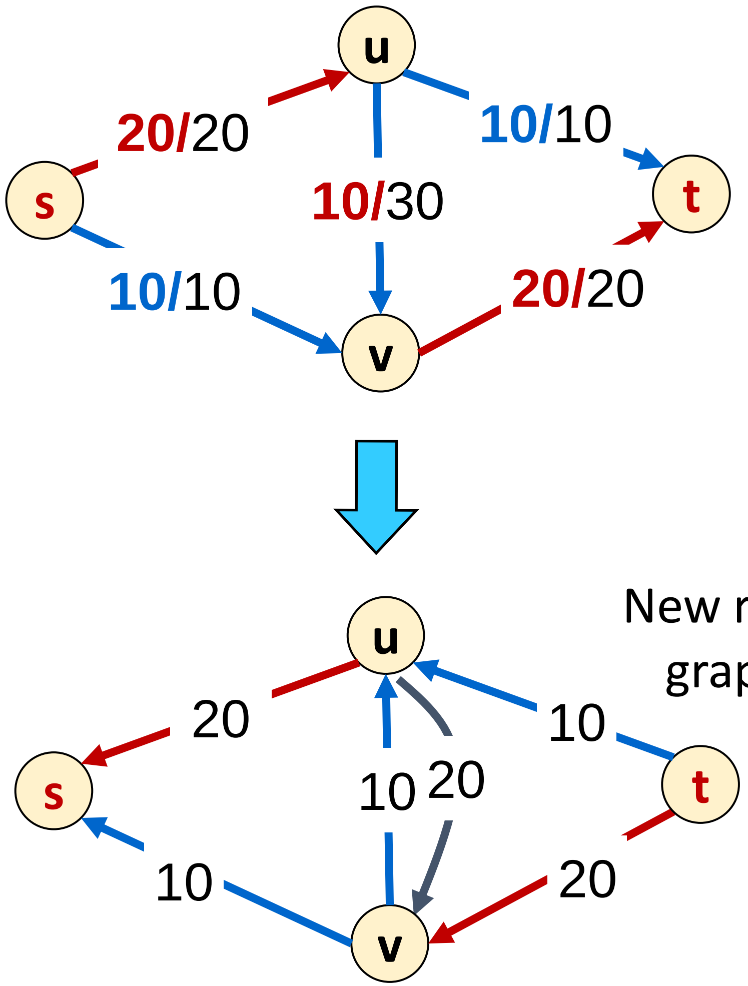
This gives us a residual graph G_f of possible changes where we draw reducing as “sending back”.



Greed Revisited: Residual Graph & Augmenting Paths



Greed Revisited: Residual Graph & Augmenting Paths



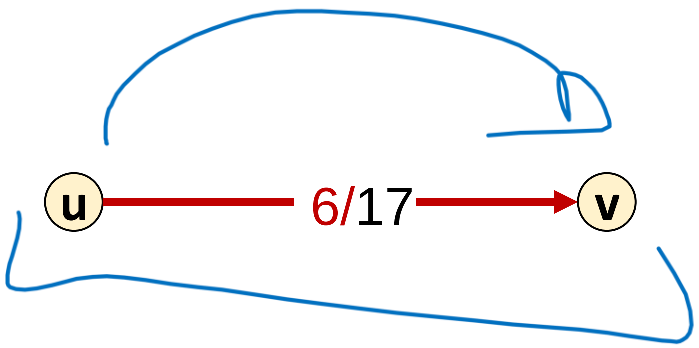
New residual graph G_f

No path can even leave s!

Residual Graphs

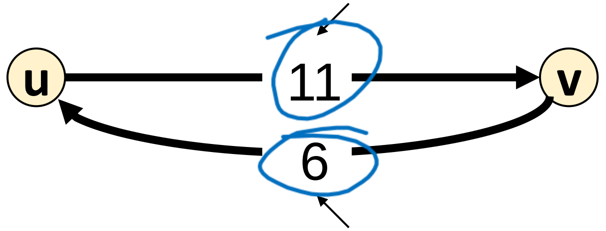
An alternative way to represent a flow network

- 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

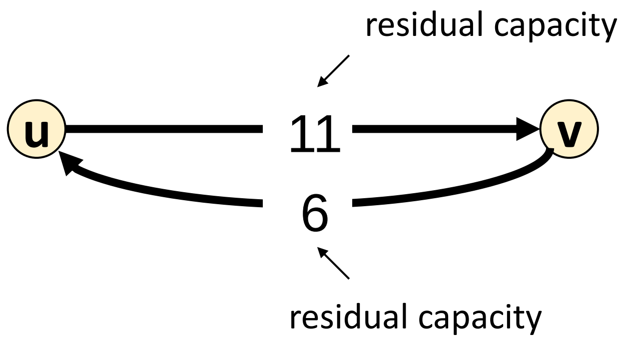
Residual graph: $G_f = (V, E_f)$.

- 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: $e^R = (v, u)$ 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 $c_f(e) > 0$.
- $E_f = \{e : f(e) < c(e)\} \cup \{e^R : f(e) > 0\}$.

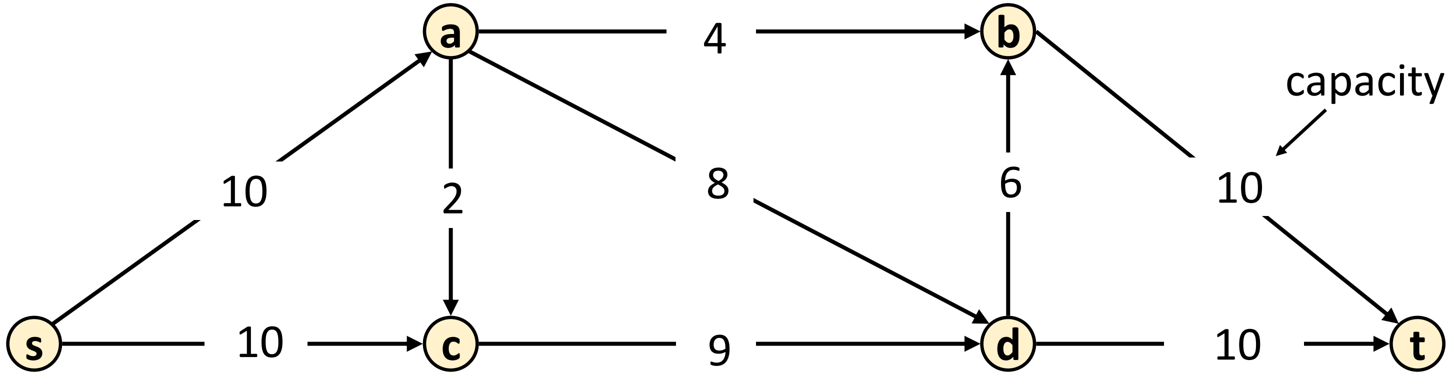
Augmenting Path: Any s-t path P in G_f. Let **bottleneck**(P)$= \min_{e \in P} c_f(e)$.

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

13

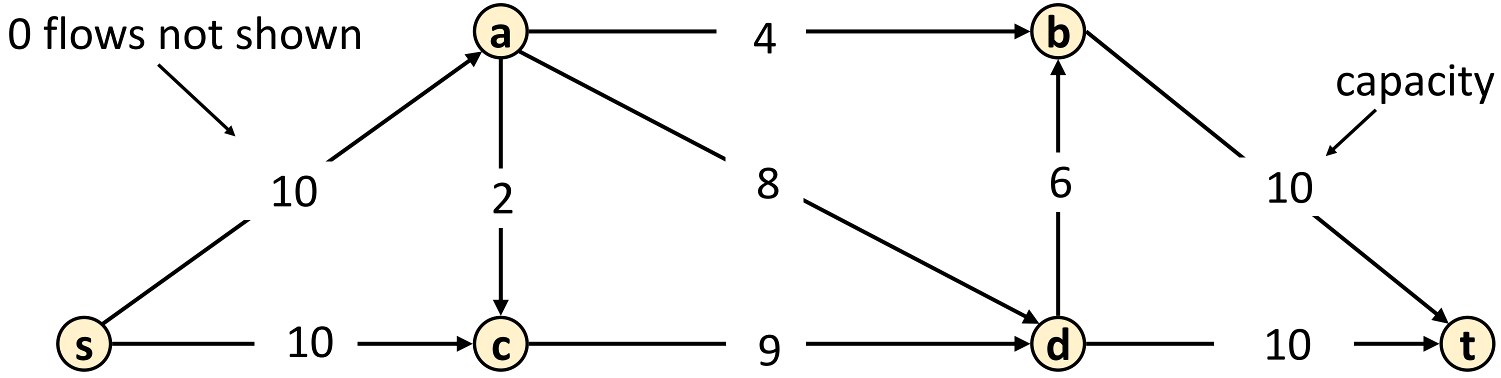
Ford-Fulkerson Algorithm

G:



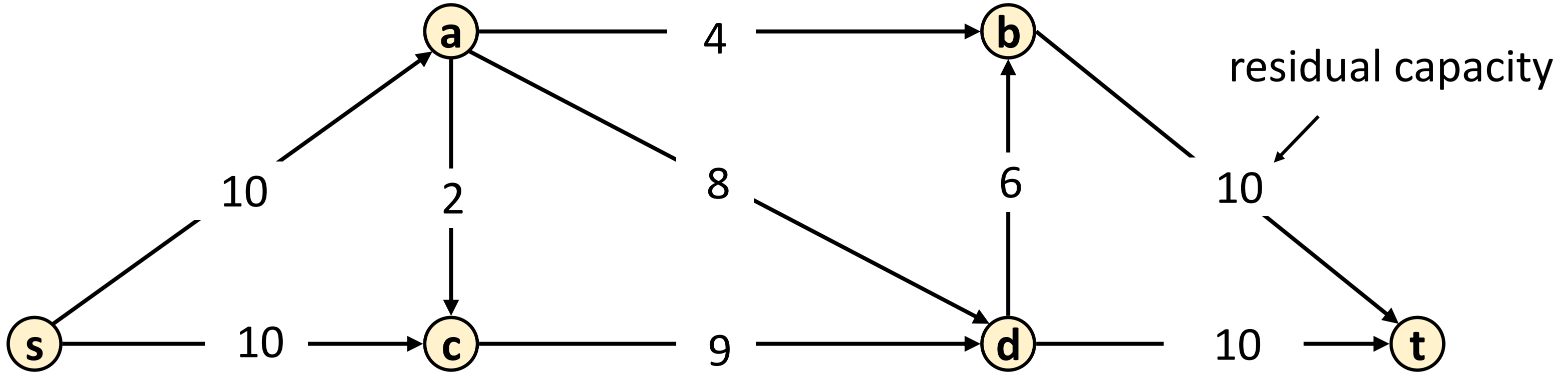
Ford-Fulkerson Algorithm

G:



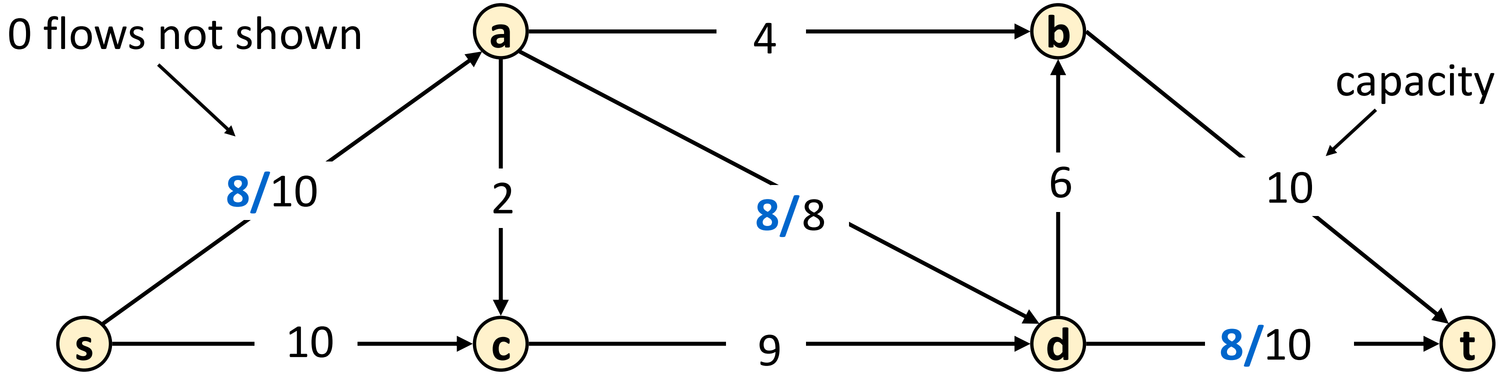
Flow value = **0**

G_f:



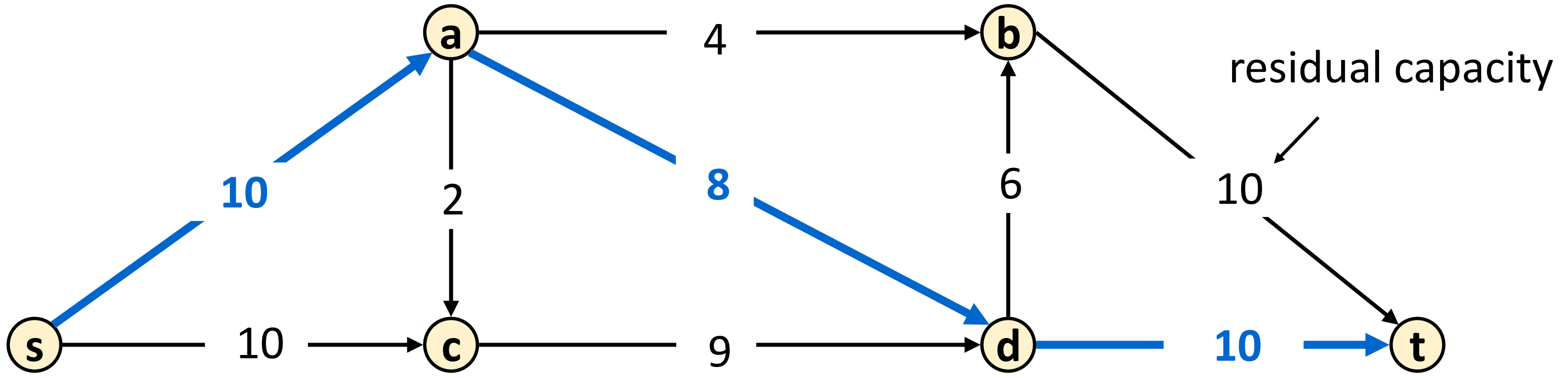
Ford-Fulkerson Algorithm

G:

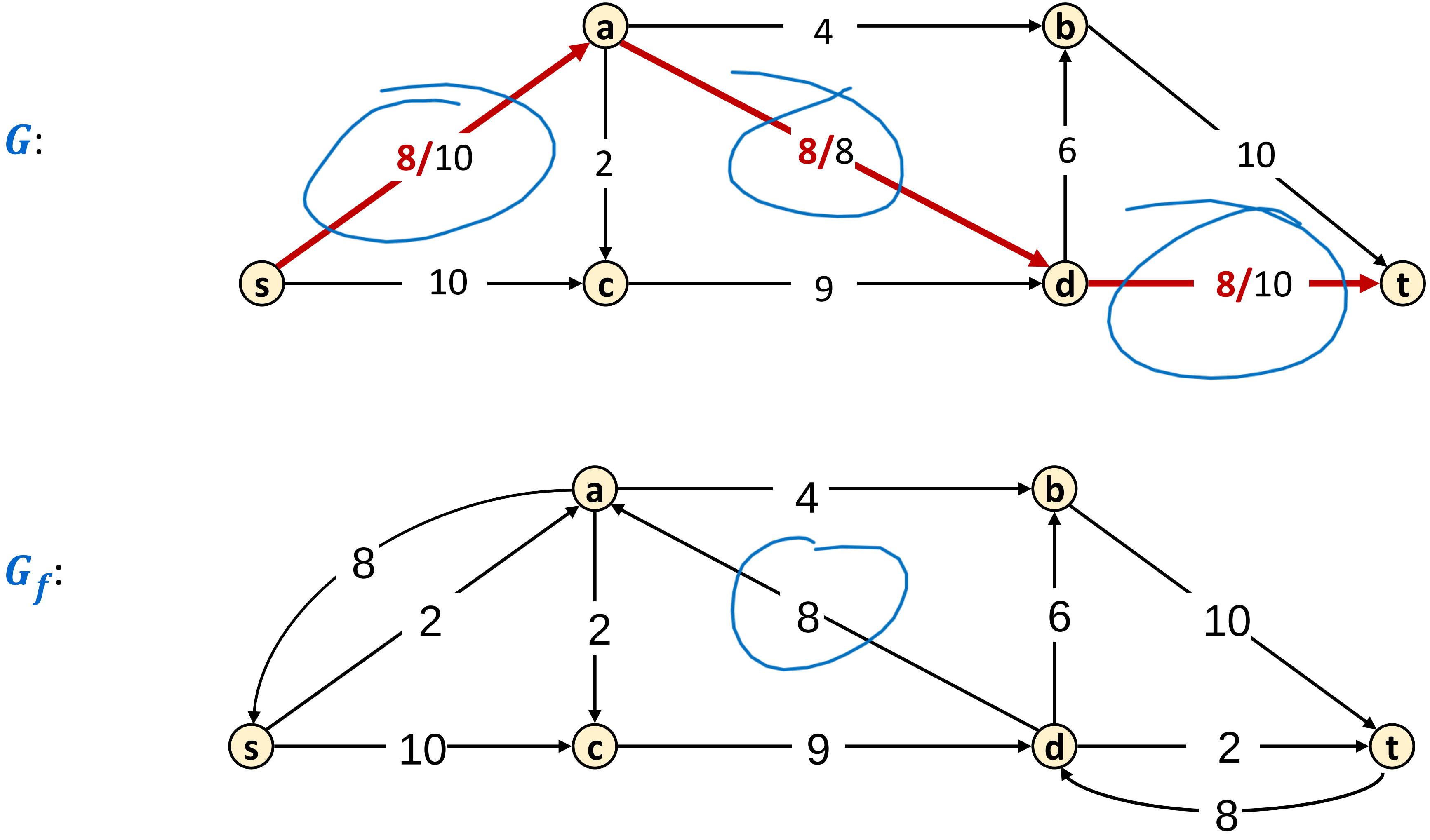


Flow value = 0
+8=8

G_f:



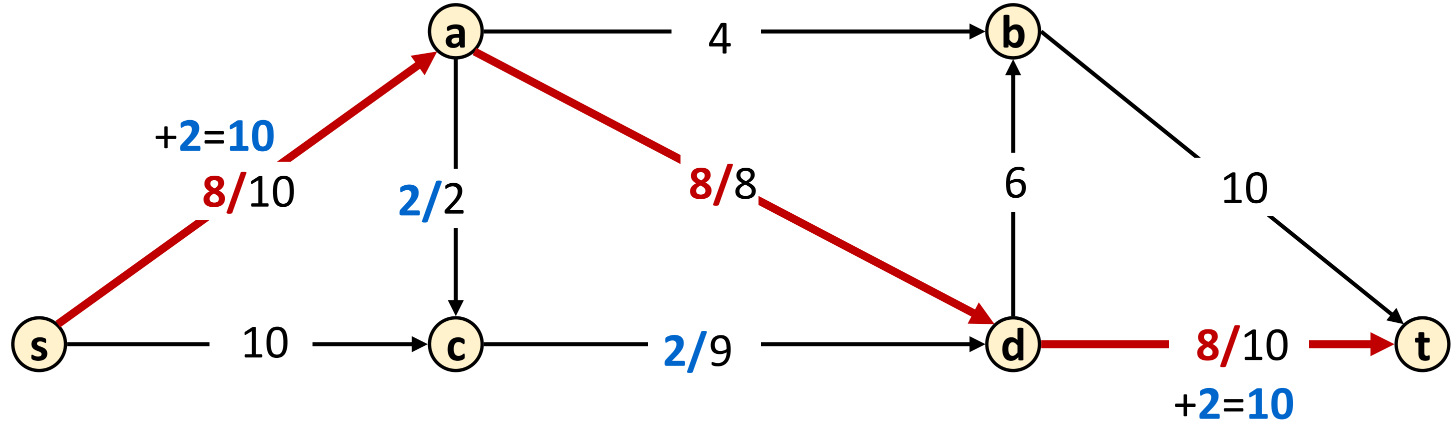
Ford-Fulkerson Algorithm



Flow value = **8**

Ford-Fulkerson Algorithm

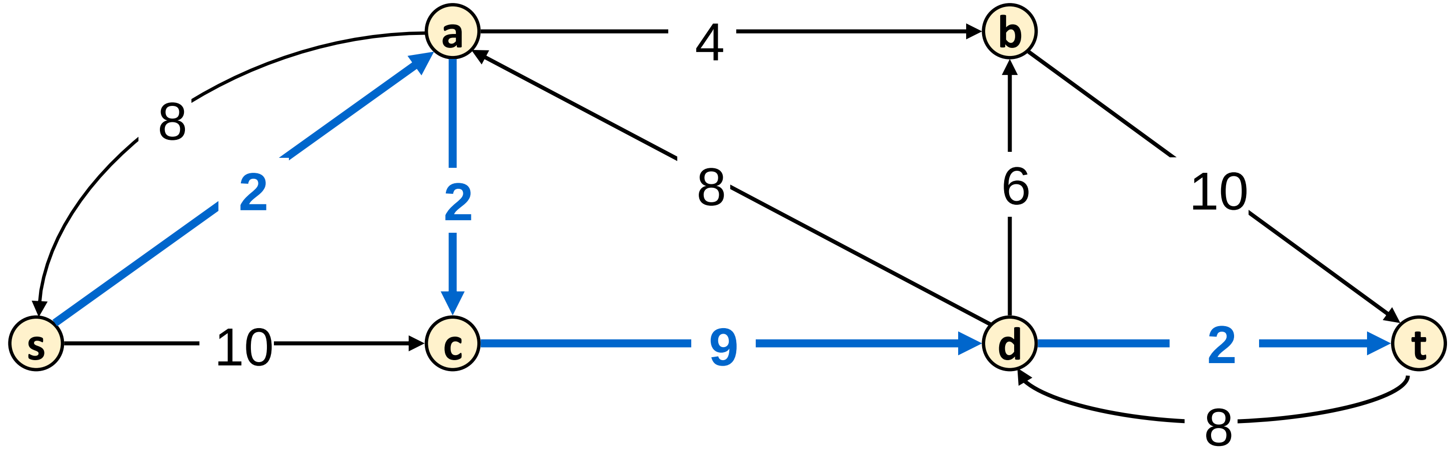
G:



Flow value = **8**

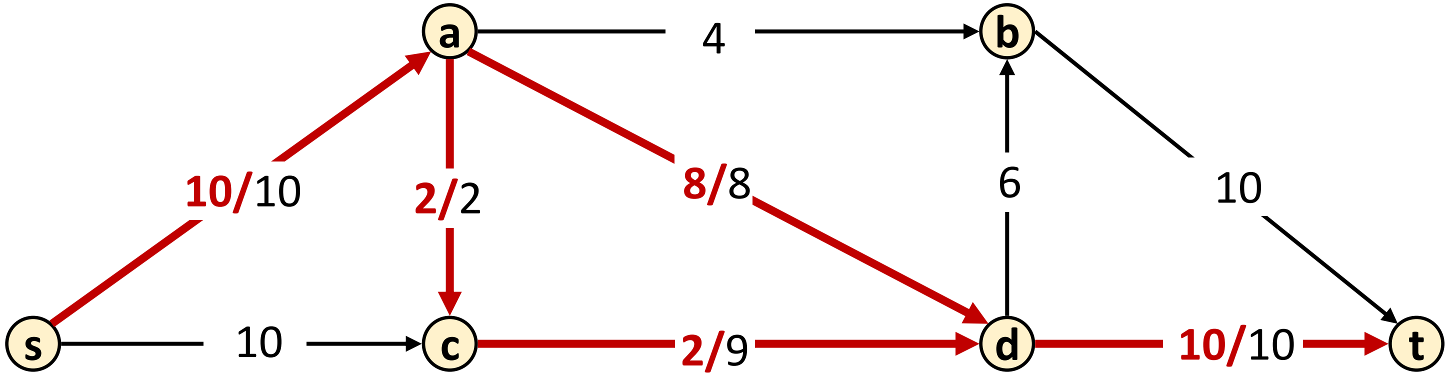
+**2**=**10**

G_f:



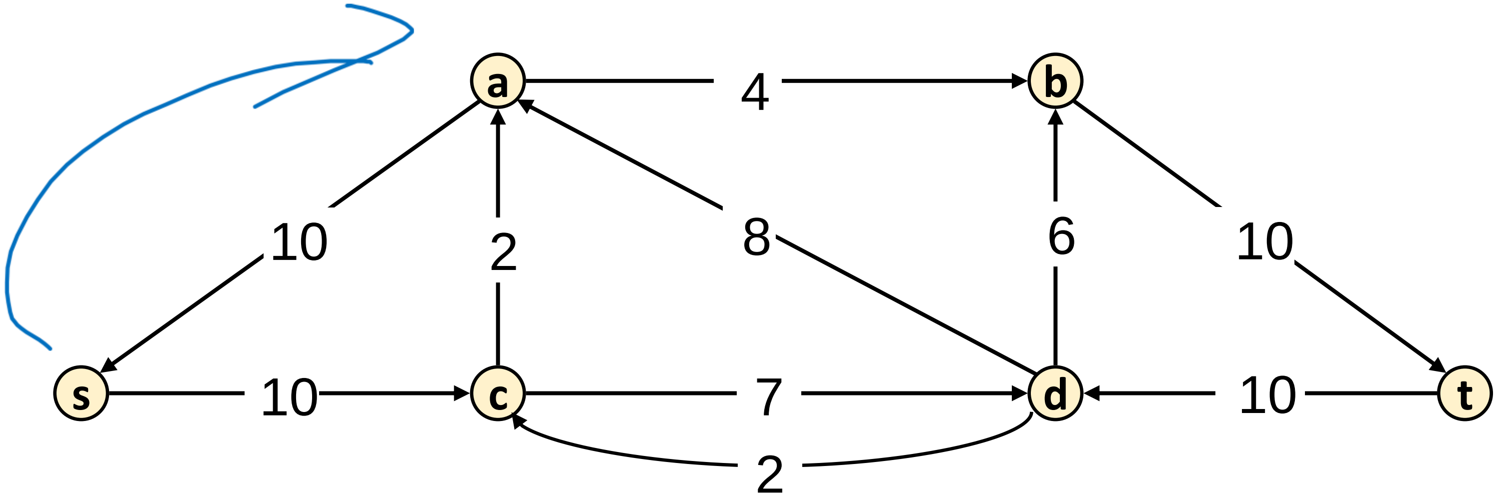
Ford-Fulkerson Algorithm

G:



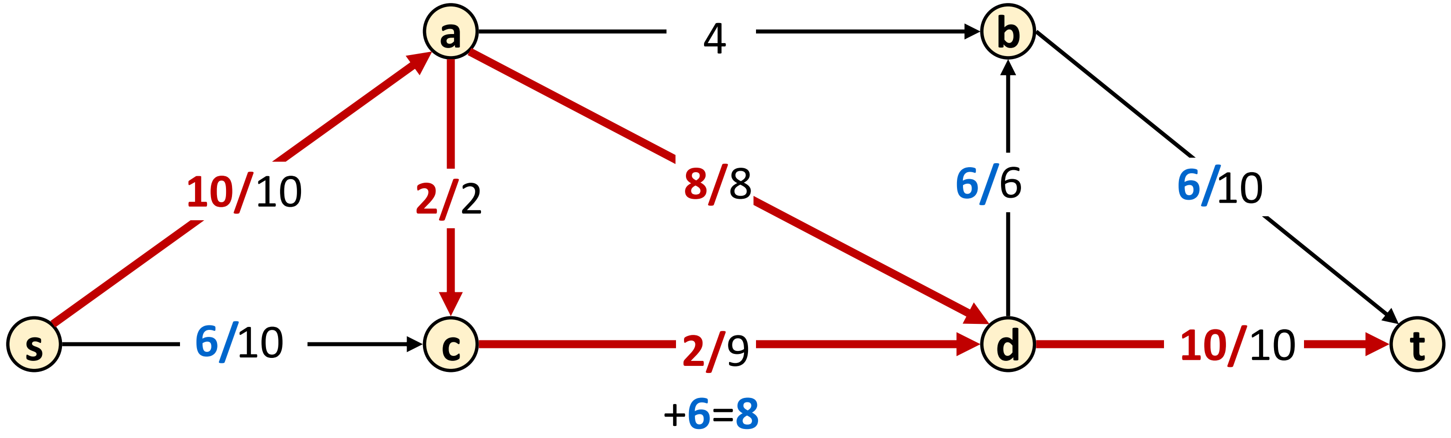
Flow value = **10**

G_f:



Ford-Fulkerson Algorithm

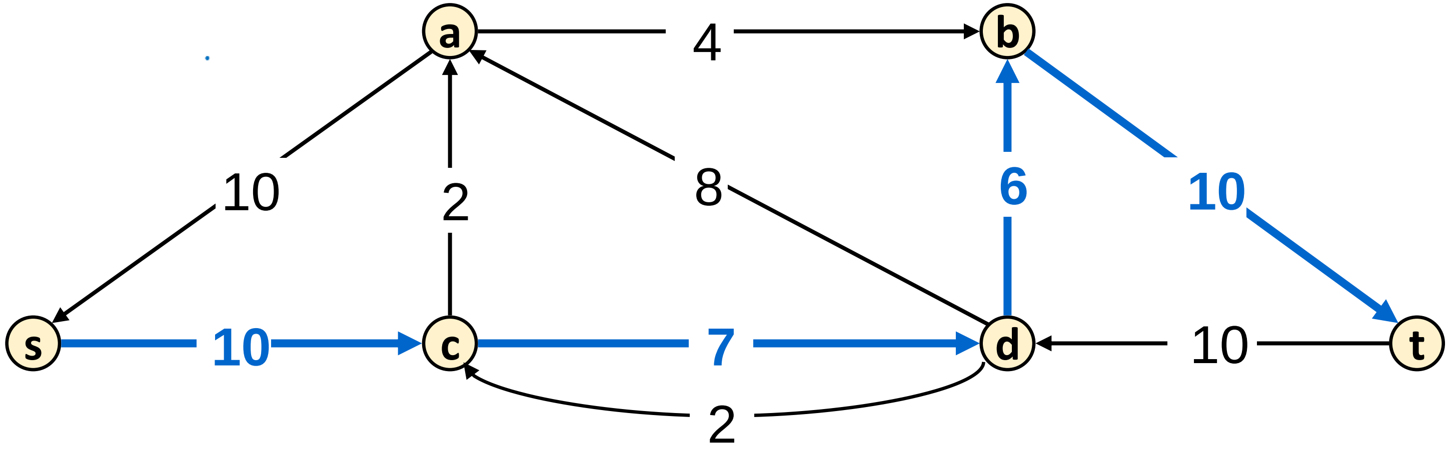
G:



Flow value = **10**

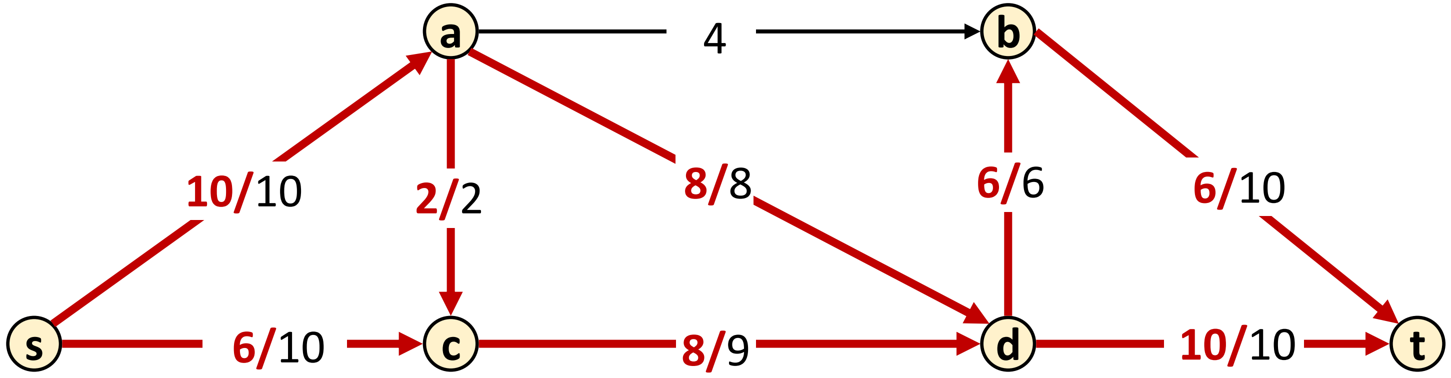
+6=16

G_f:



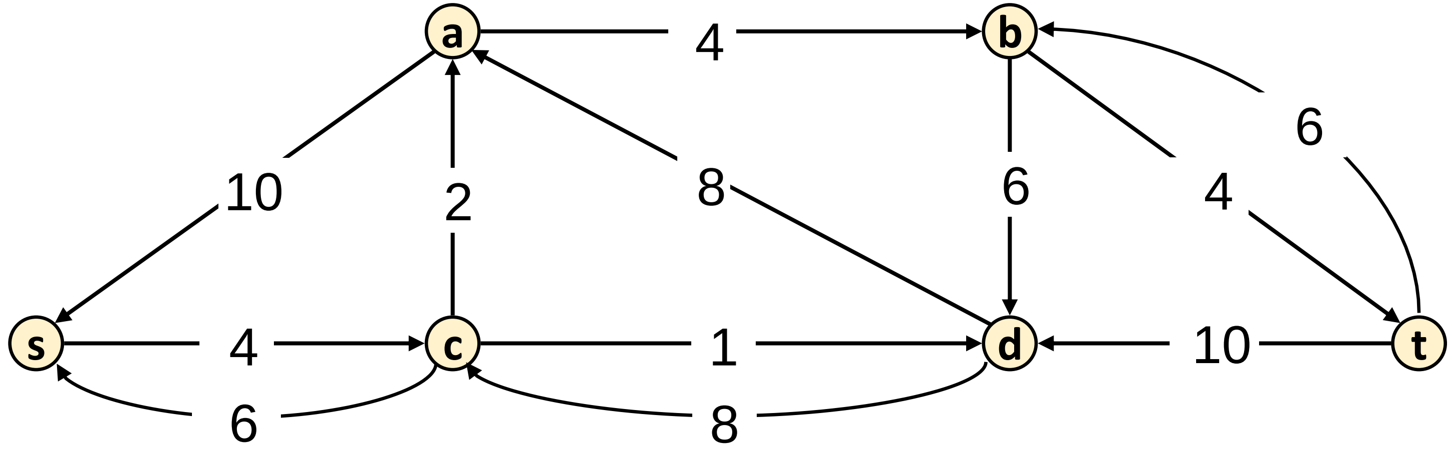
Ford-Fulkerson Algorithm

G:



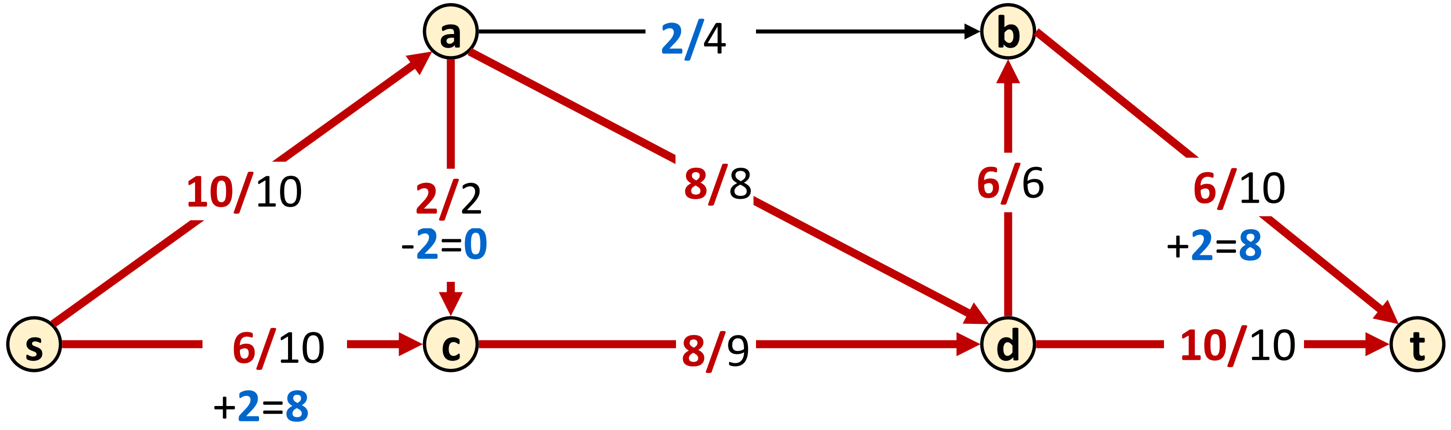
Flow value = **16**

G_f:



Ford-Fulkerson Algorithm

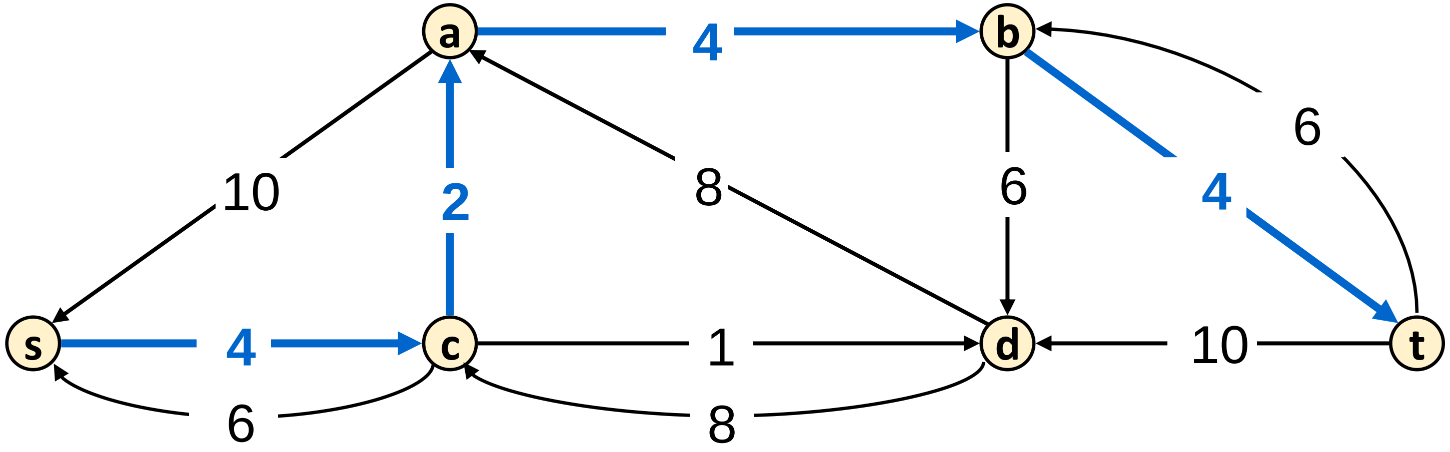
G:



Flow value = **16**

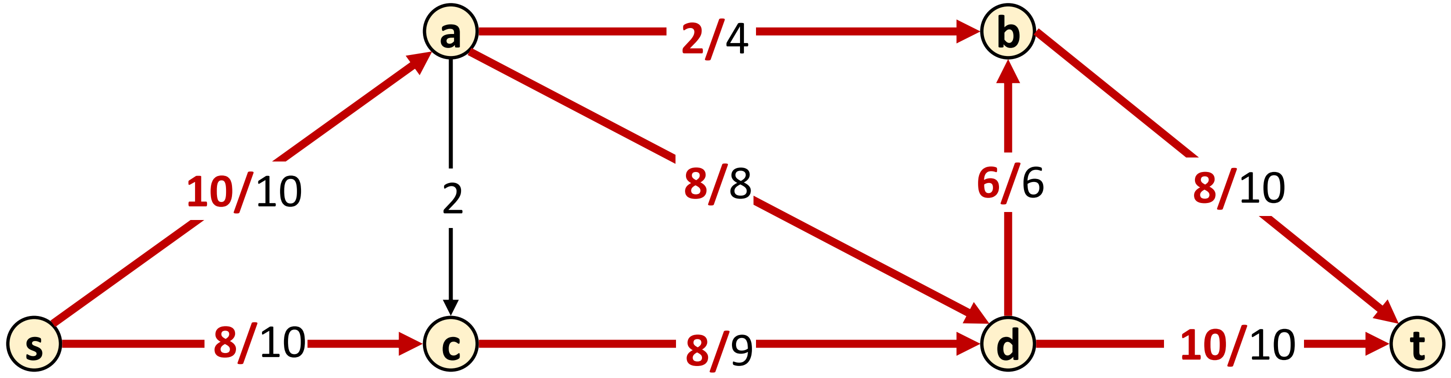
+2=**18**

G_f:



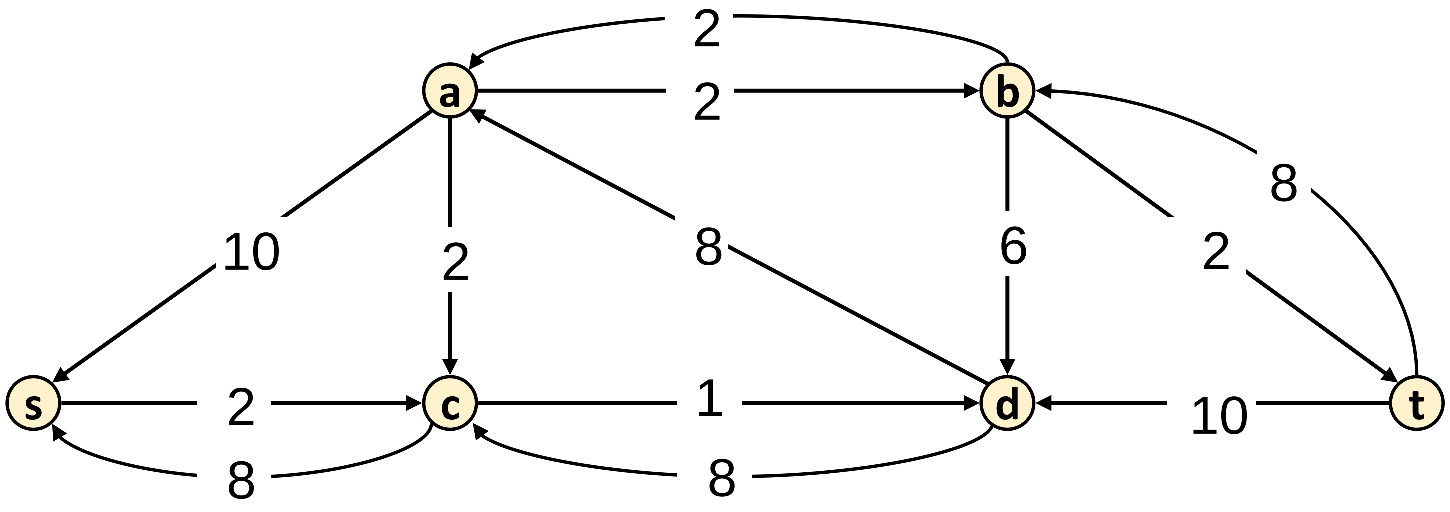
Ford-Fulkerson Algorithm

G:

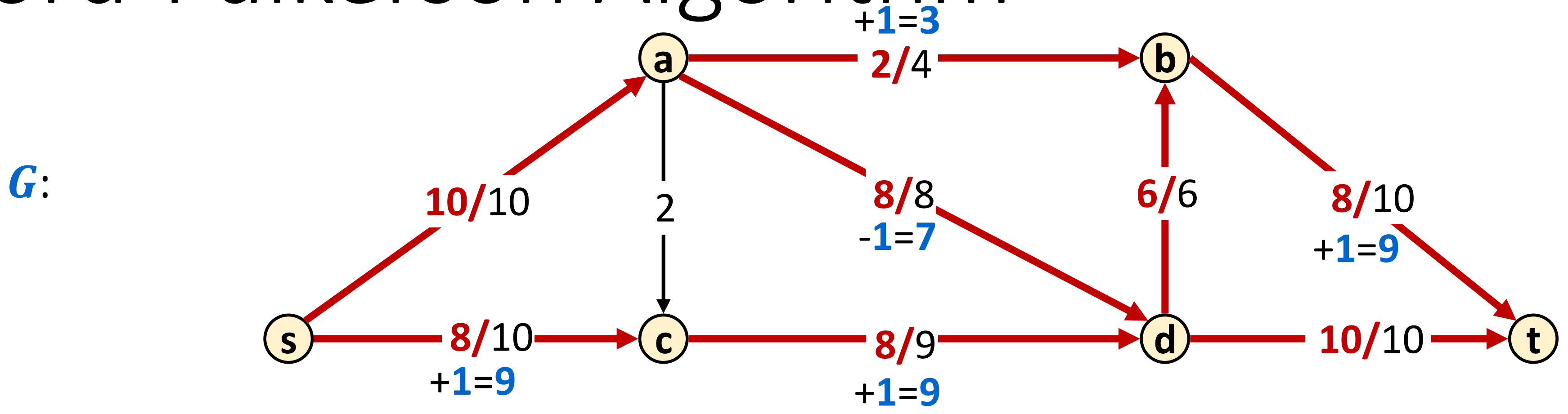


Flow value = **18**

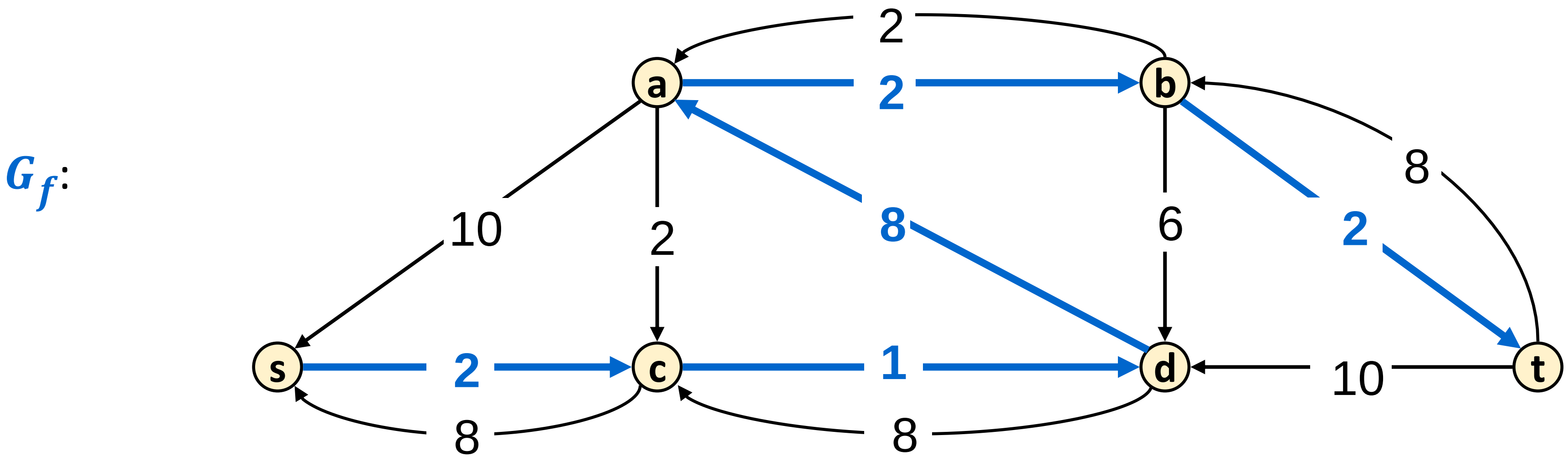
G_f:



Ford-Fulkerson Algorithm

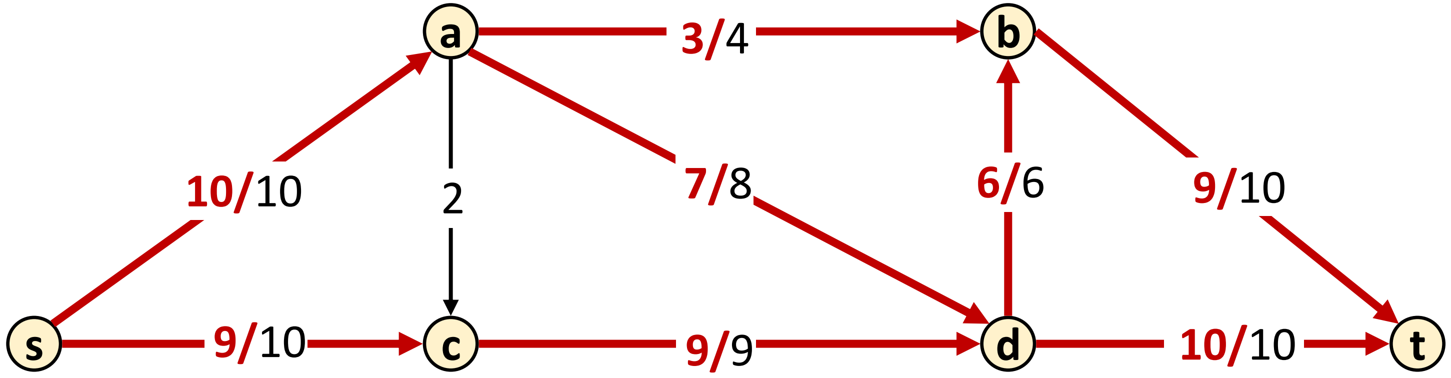


Flow value = 18
+1=19



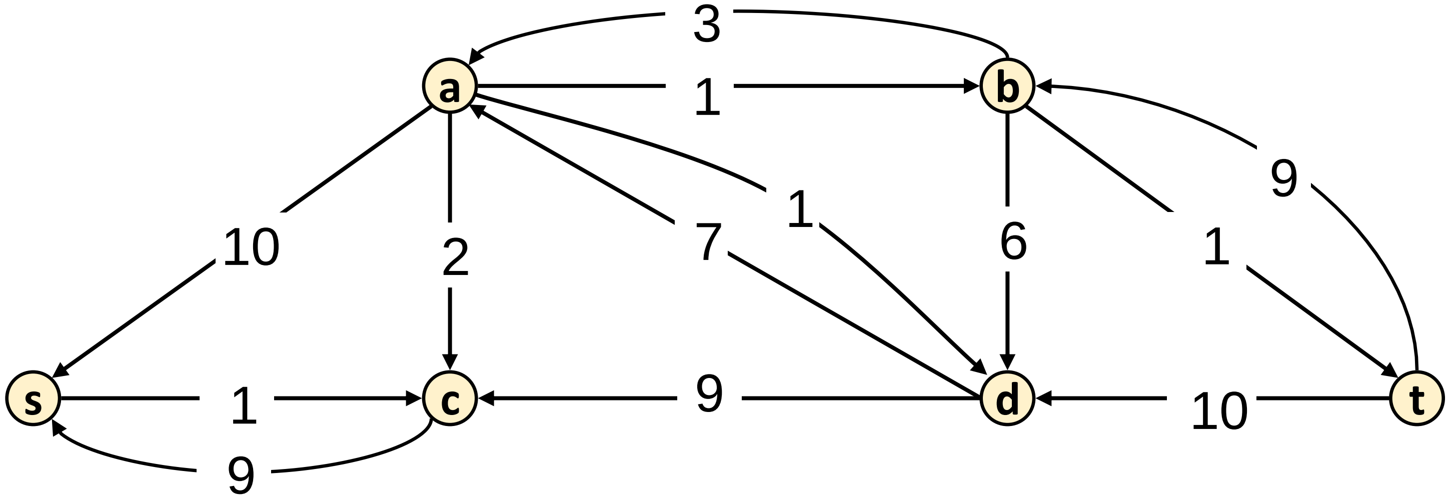
Ford-Fulkerson Algorithm

G:



Flow value = **19**

G_f:



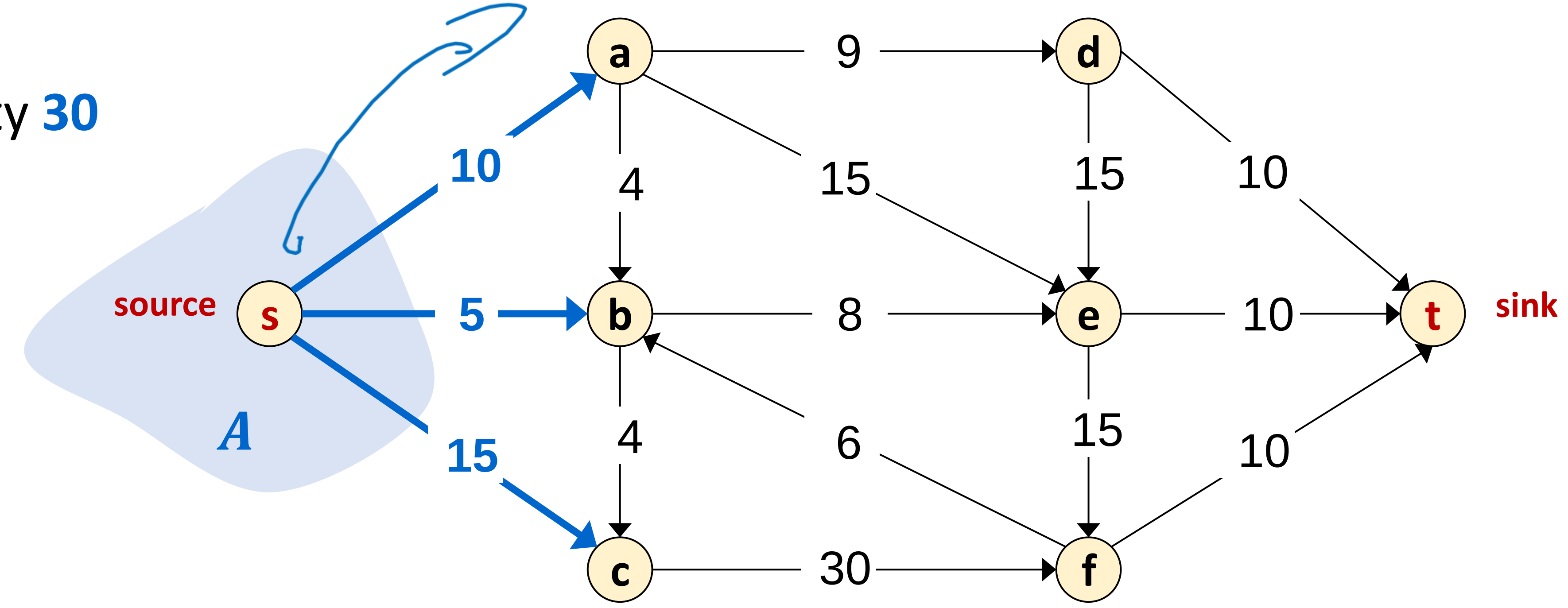
Cuts

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**



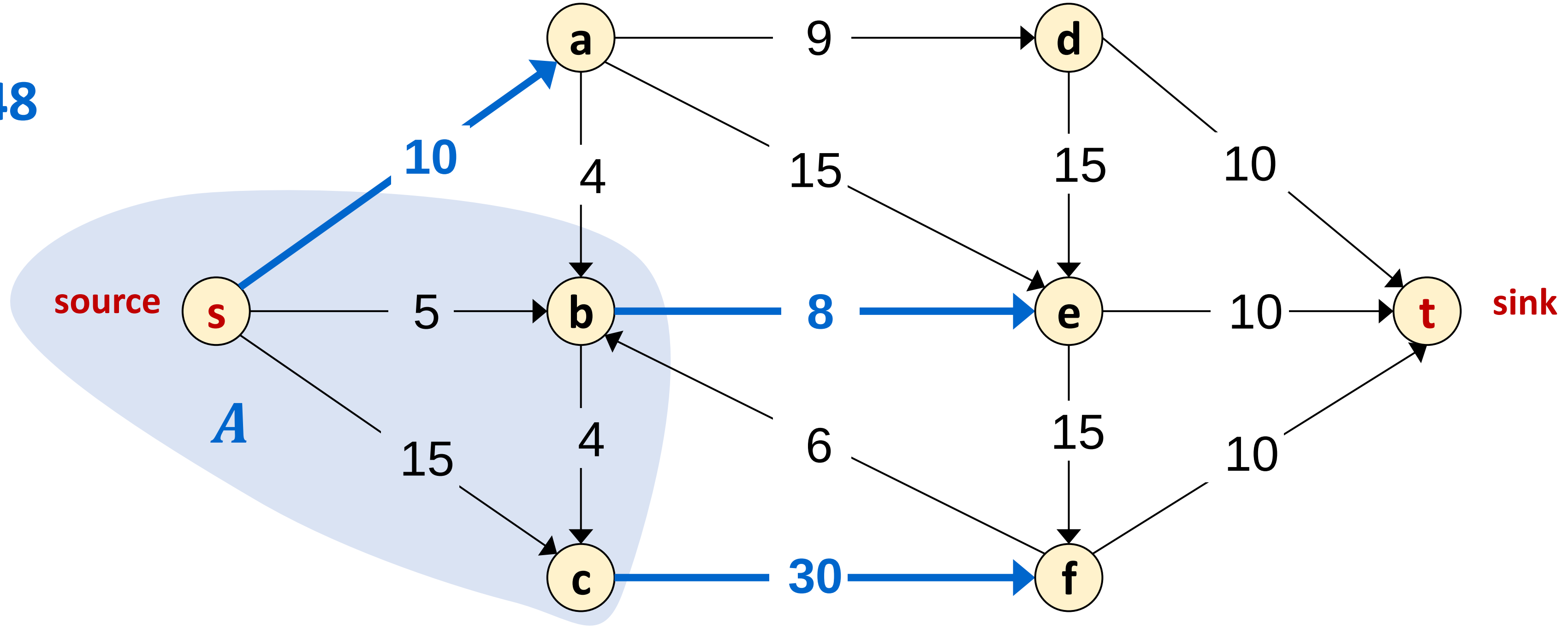
Cuts

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**



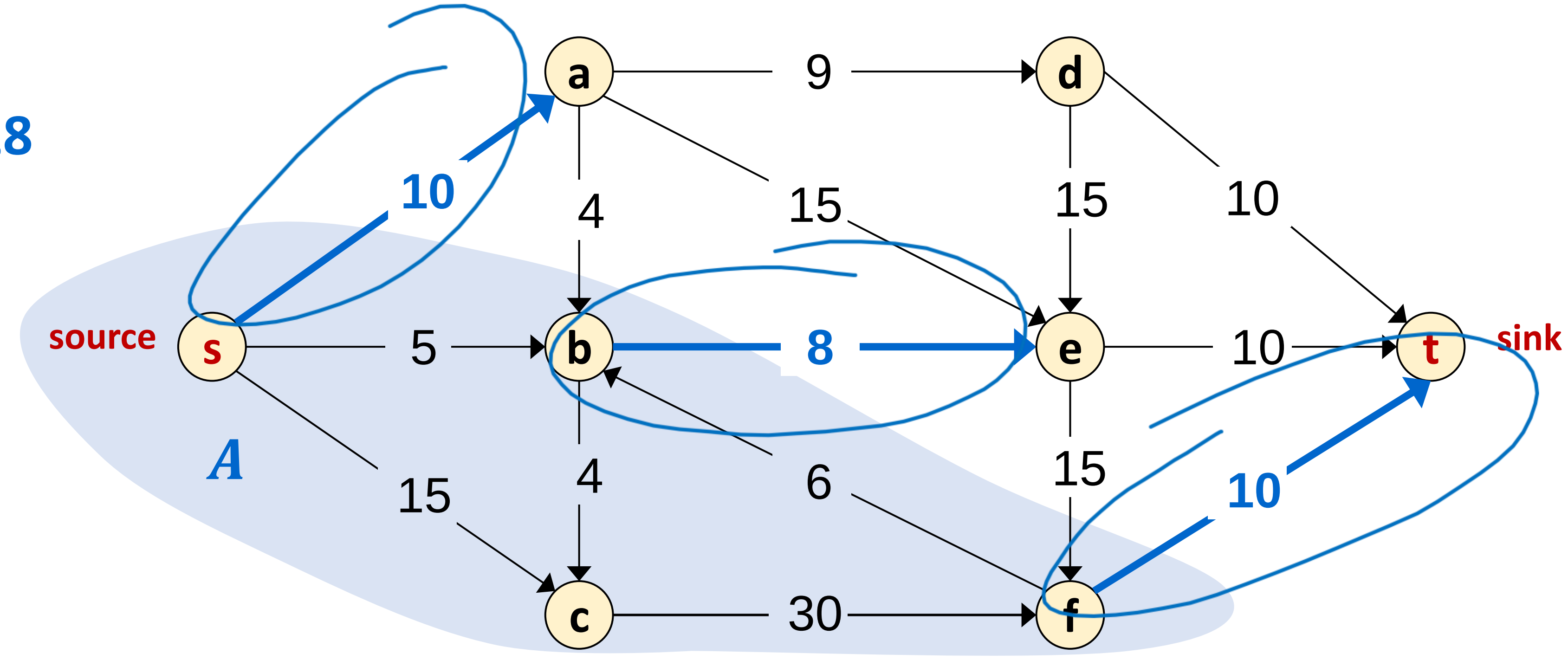
Minimum Cut Problem

Minimum s-t cut problem:

Given: a flow network

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

capacity **28**



Flows and Cuts

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

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

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., $v(f) \leq c(A, B)$.

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

Corollary: If $v(f) = c(A, B)$ then f is a maximum flow and (A, 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.

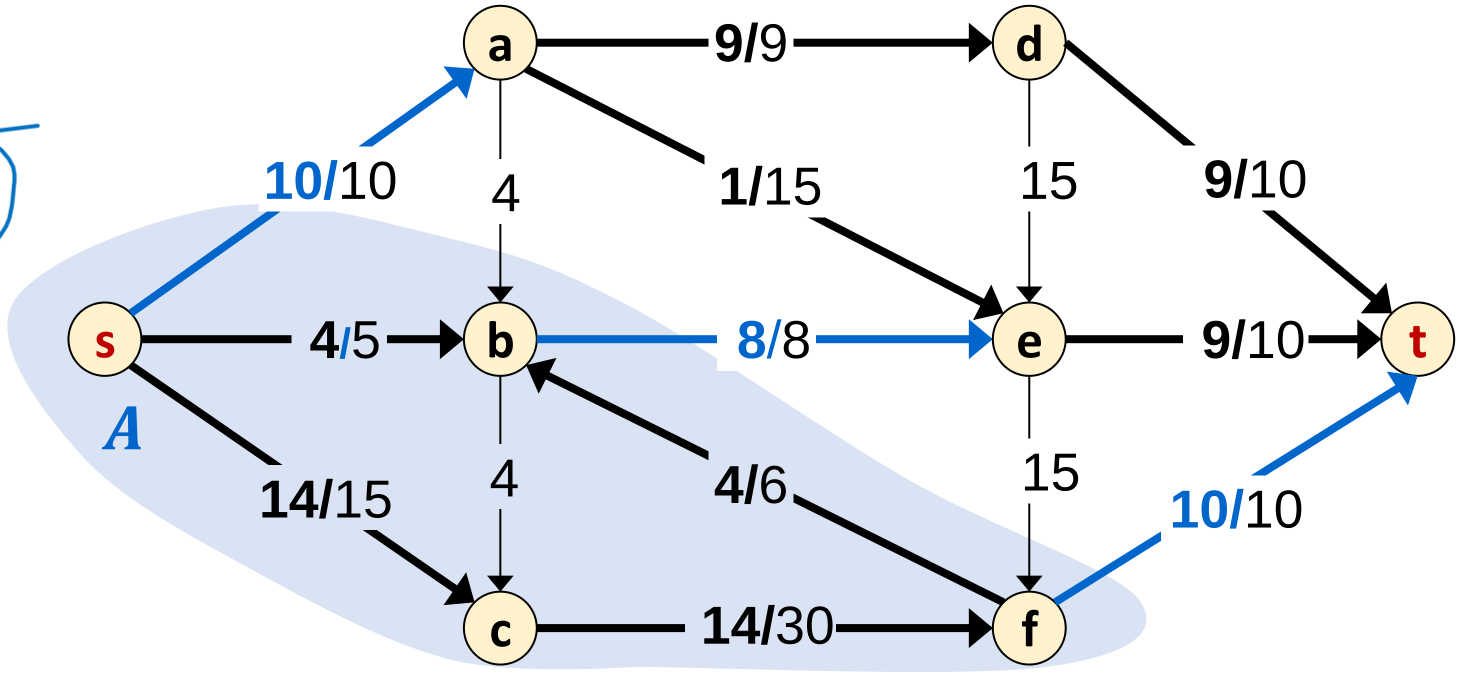
Certificate of Optimality

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!

Max-Flow Min-Cut Theorem

Augmenting Path Theorem: Flow f is a max flow ⇔ there are no augmenting paths wrt f

Max-Flow Min-Cut Theorem: The value of the max flow equals the value of the min cut.

[Elias-Feinstein-Shannon 1956, Ford-Fulkerson 1956]

"Maxflow = Mincut"

Proof: We prove both together by showing that all of these are equivalent:

(i) There is a cut (A, B) such that $v(f) = c(A, B)$.

(ii) Flow f is a max flow.

(iii) There is no augmenting path w.r.t. f.

(i) ⇒ (ii): Comes from weak duality lemma.

(ii) ⇒ (iii): (by contradiction)

If there is an augmenting path w.r.t. flow f then we can improve f. Therefore f is not a max flow.

(iii) ⇒ (i): We will use the residual graph to identify a cut whose capacity matches the flow

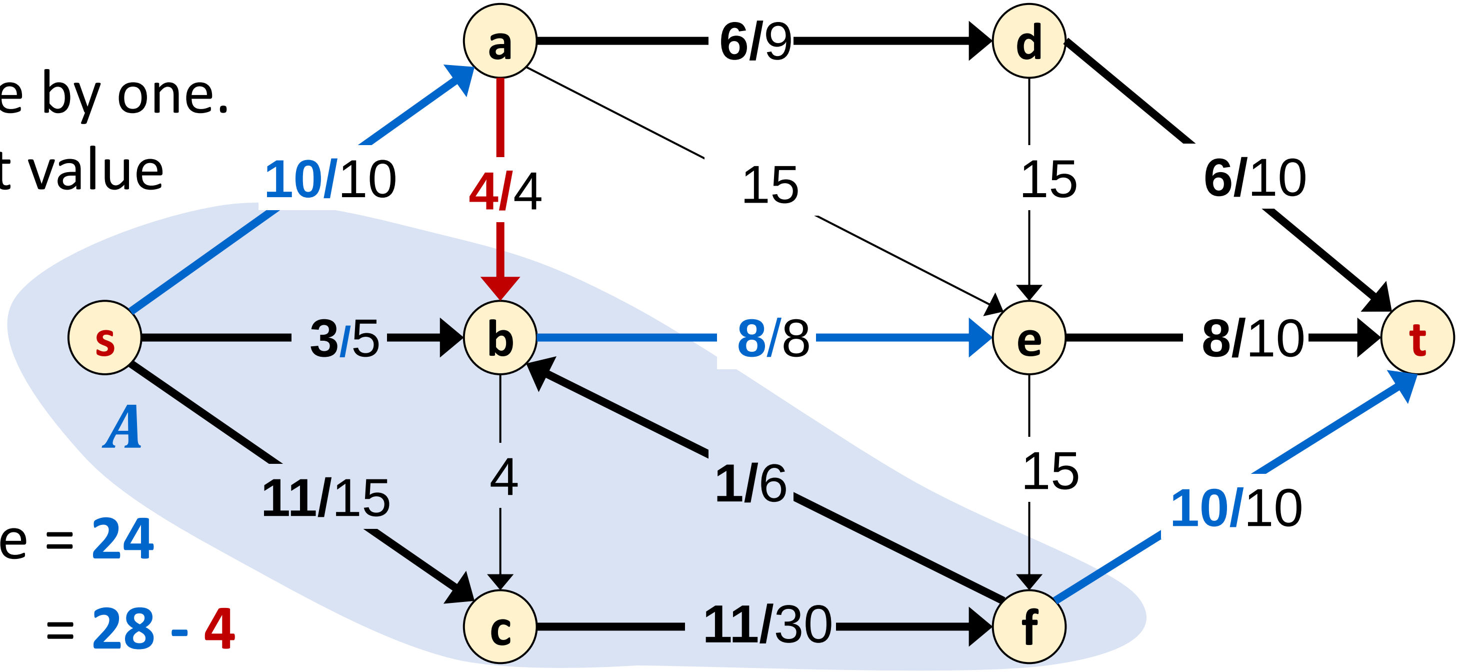
Flow Value Lemma – Idea

Flow Value Lemma: Let f be any s-t flow and (A, B) be any s-t cut. The net value of the flow sent across the cut equals $v(f)$:

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

Why is it true?

- Add vertices to s side one by one.
- By flow conservation, net value doesn't change



value = **24**

= **28 - 4**

Flow Value Lemma – Proof

Flow Value Lemma: Let f be any s-t flow and (A, B) be any s-t cut. The net value of the flow sent across the cut equals $v(f)$:

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

Proof:

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

$$= \sum_{e \text{ out of } s} f(e) - \sum_{e \text{ into } s} f(e) + \sum_{v \in A - \{s\}} \left[\sum_{e \text{ out of } v} f(e) - \sum_{e \text{ into } v} f(e) \right]$$

$$= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e)$$

$= \mathbf{0}$. No edges into s since it is a source

$= \mathbf{0}$ by flow conservation.

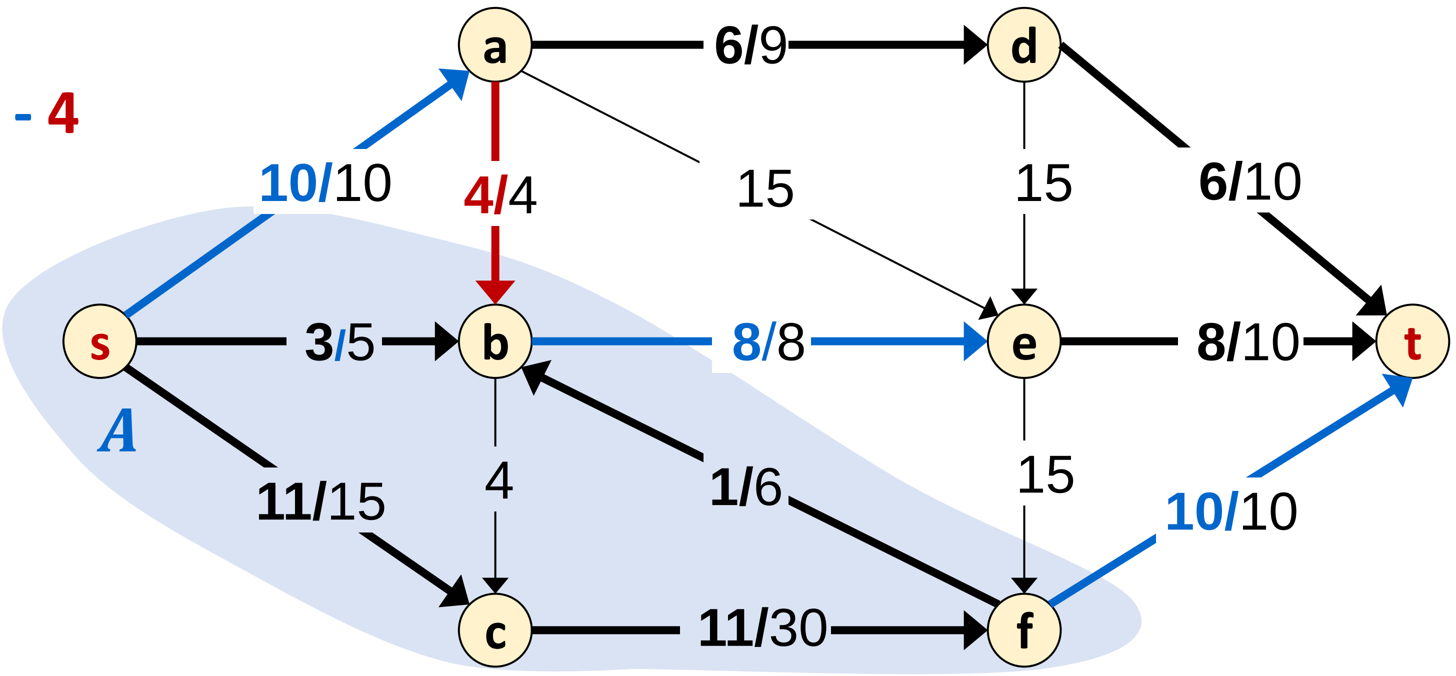
Contributions from internal edges of A cancel.

Weak Duality - Idea

Weak Duality: Let f be any s-t flow and (A, B) be any s-t cut. The value of the flow is at most the capacity of the cut; i.e., $v(f) \leq c(A, B)$:

Value of flow = **24** = **28** - **4**

Capacity of cut = **28**



Weak Duality - Proof

Weak Duality: Let f be any s-t flow and (A, B) be any s-t cut. The value of the flow is at most the capacity of the cut; i.e., $v(f) \leq c(A, B)$.

Proof:

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

$$\leq \sum_{e \text{ out of } A} f(e) \qquad \text{since } f(e) \geq 0$$

$$\leq \sum_{e \text{ out of } A} c(e) \qquad \text{since } f(e) \leq c(e)$$

$$= c(A, B)$$

■

Proof of Max-Flow Min-Cut Theorem

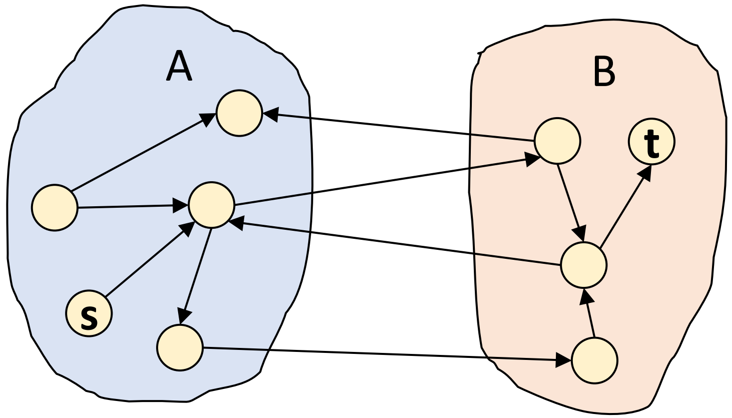
(iii) ⇒ (i):

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

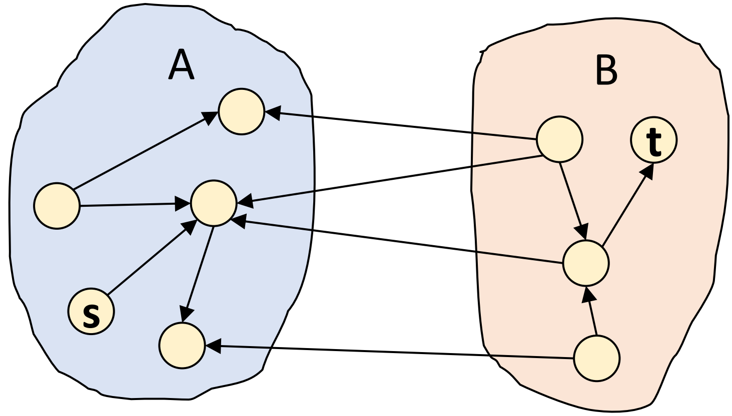
Proof of Claim: Let f be a flow with no augmenting paths.

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), $t \notin A$.



original network



residual graph

Proof: Identifying the Min Cut

(iii) ⇒ (i):

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

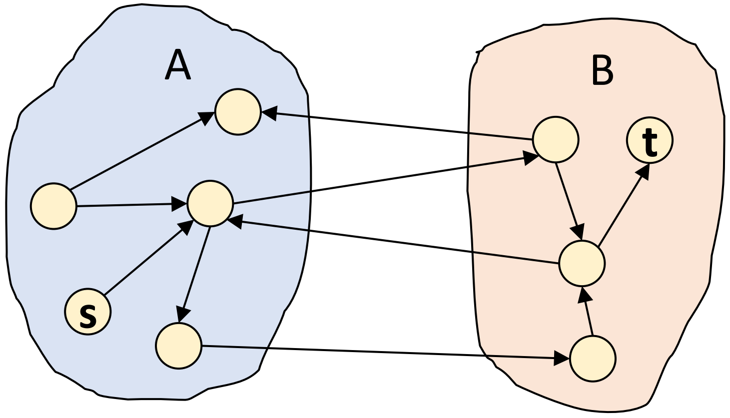
Proof of Claim: Let f be a flow with no augmenting paths.

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

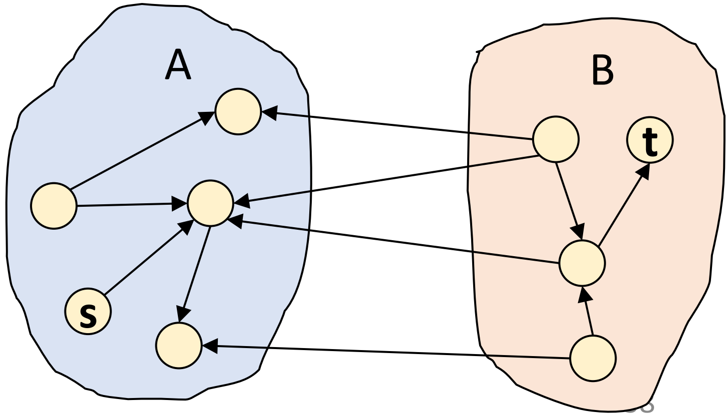
- By definition of A, $s \in A$.
- Since no augmenting path (s-t path in G_f), $t \notin A$.

Then

$$v(f) = \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e)$$ (by Flow-Value Lemma)



original network



residual graph

Identifying the Min Cut: No Inflow

<u>(iii) ⇒ (i):</u>

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

Proof of Claim: Let f be a flow with no augmenting paths.

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

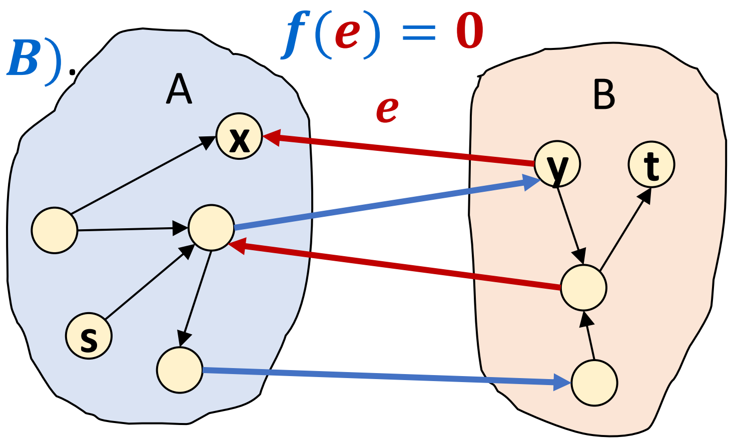
- By definition of A, $s \in A$.
- Since no augmenting path (s-t path in G_f), $t \notin A$.

Then

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

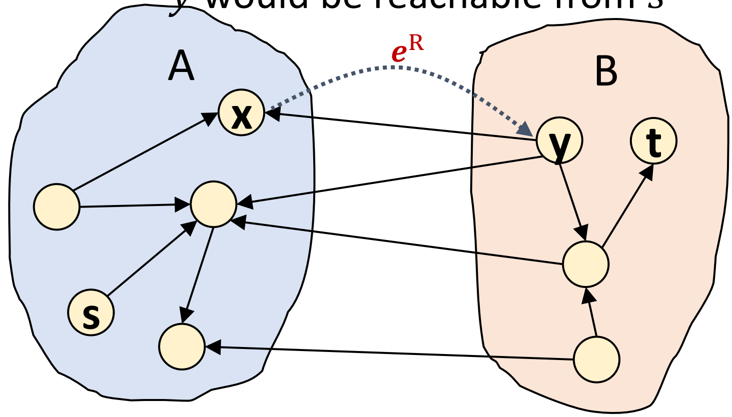
$$= \sum_{e \text{ out of } A} f(e)$$

(**By contradiction:** If an edge going into A had flow then the backward edge would be in the residual graph, so the edge should not cross the cut)



original network

e^R can't exist because then y would be reachable from s



residual graph

Identifying the Min Cut: Saturated Outflow

(iii) ⇒ (i):

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

Proof of Claim: 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), $t \notin A$.

Then

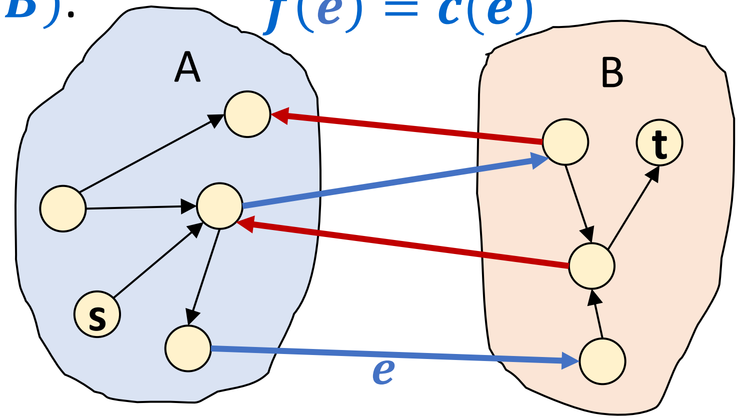
$$v(f) = \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e)$$

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

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

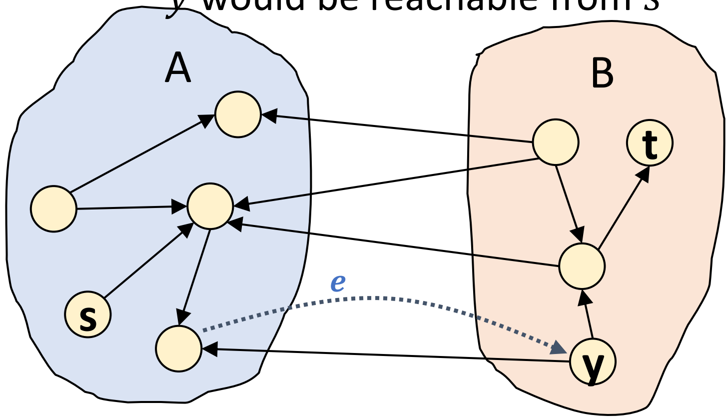
(**By contradiction:** If an edge going out of A had unused capacity then the forward edge would be in the residual graph, so the edge should not cross the cut)

"e is saturated"
No unused capacity on e
$f(e) = c(e)$



original network

e^R can't exist because then y would be reachable from s



residual graph

Identifying the Min Cut: Conclusion

(iii) ⇒ (i):

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

Proof of Claim: Let f be a flow with no augmenting paths.

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

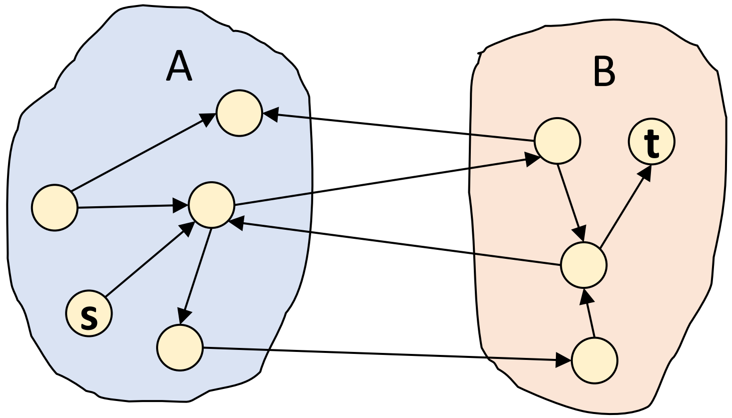
- By definition of A, $s \in A$.
- Since no augmenting path (s-t path in G_f), $t \notin A$.

Then

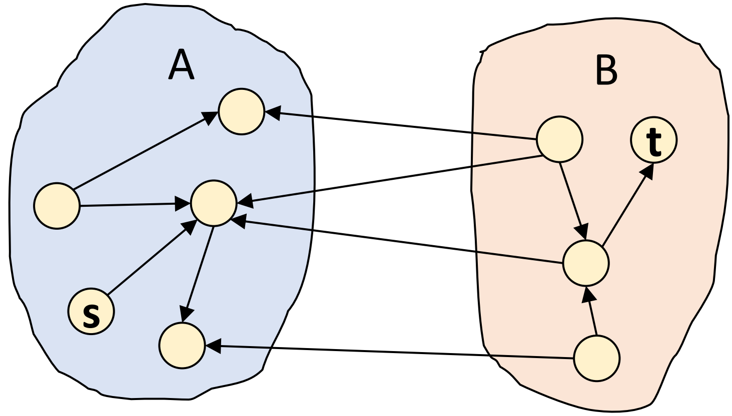
$$v(f) = \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e)$$

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

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



original network



residual graph