# CS7800: Advanced Algorithms

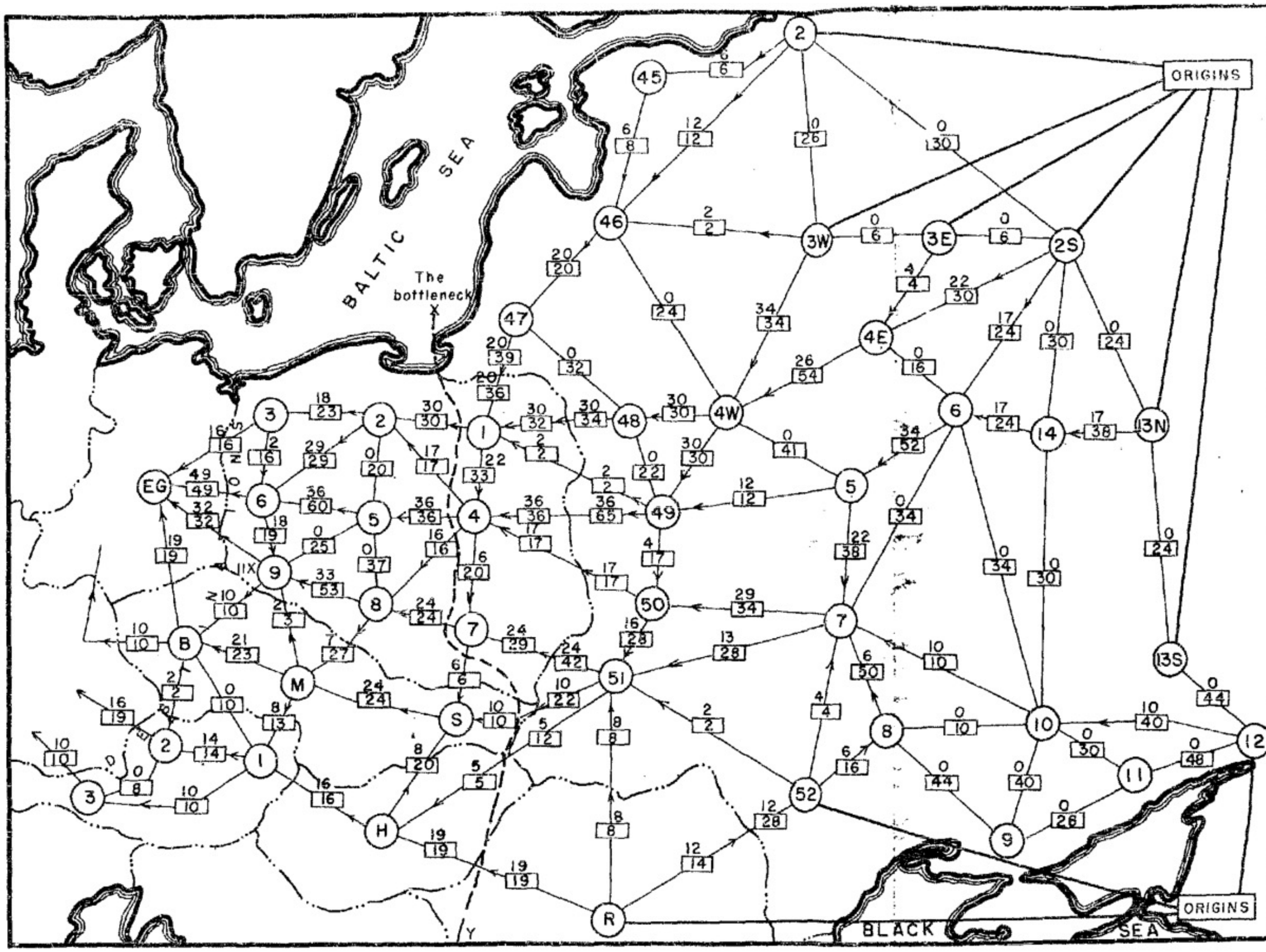
Network Flow I

- Ford Fulkerson
- Duality

Jonathan Ullman

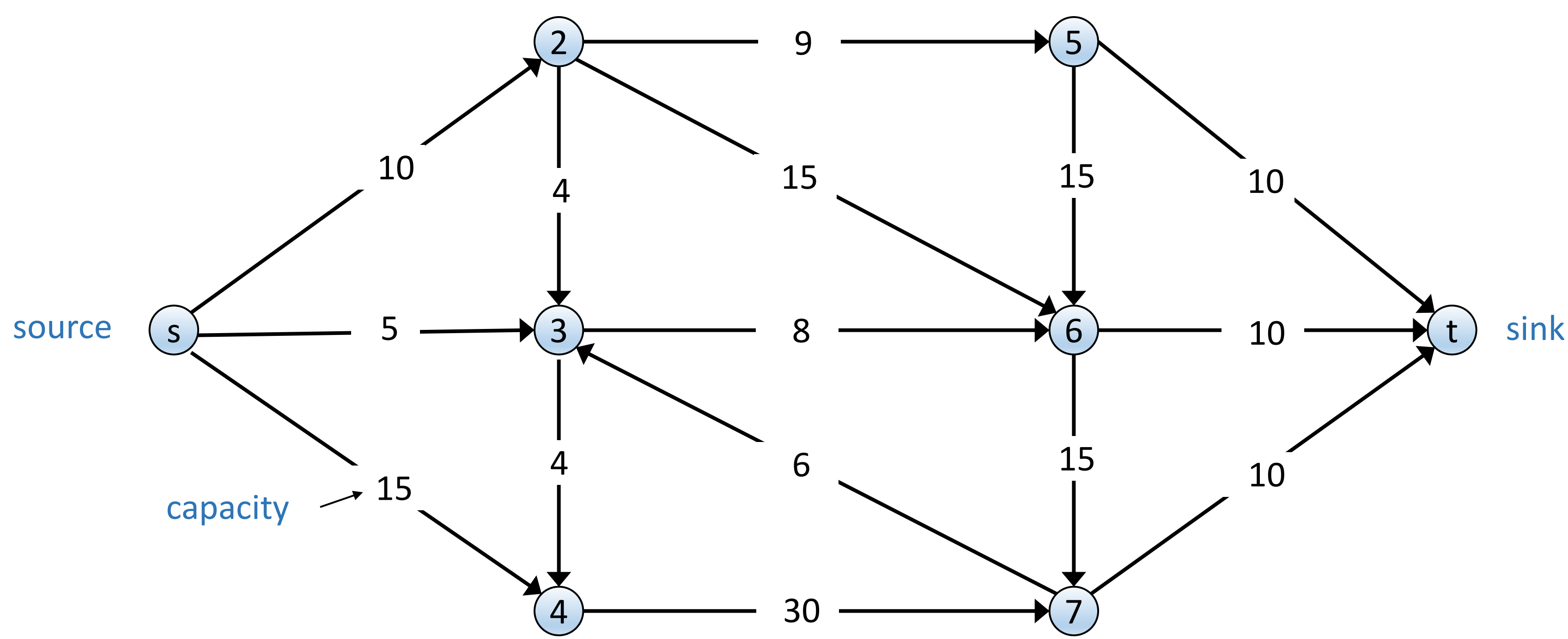
09-27-22

# Flows and Cuts



# Flow Networks

- Directed graph $G = (V, E)$
- Two special nodes: source $s$ and sink $t$
- Edge capacities $c(e)$

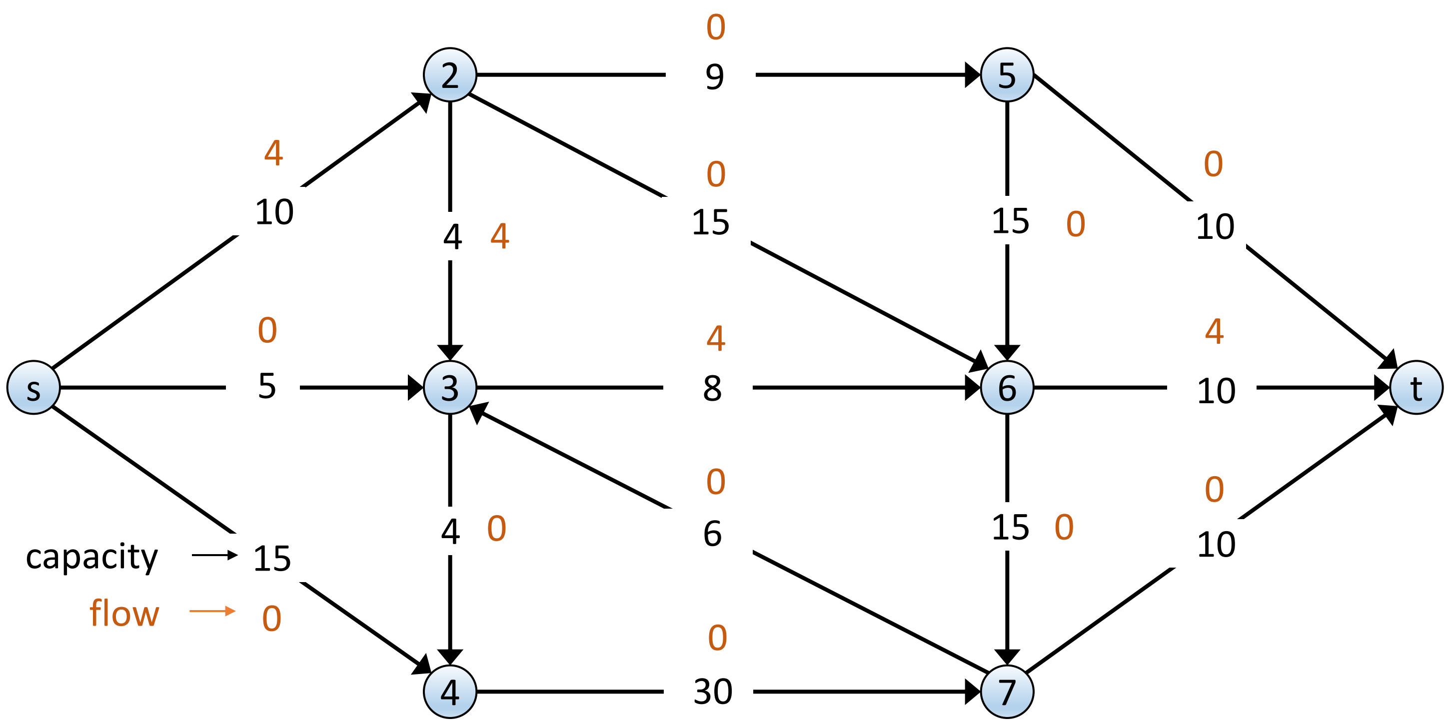


# Flows

- An s-t flow is a function $f(e)$ such that
  - For every $e \in E$, $0 \leq f(e) \leq c(e)$ (non-negativity, capacity)
  - For every $v \in E$, $\sum_{e \text{ in to } v} f(e) = \sum_{e \text{ out of } v} f(e)$ (flow conservation)

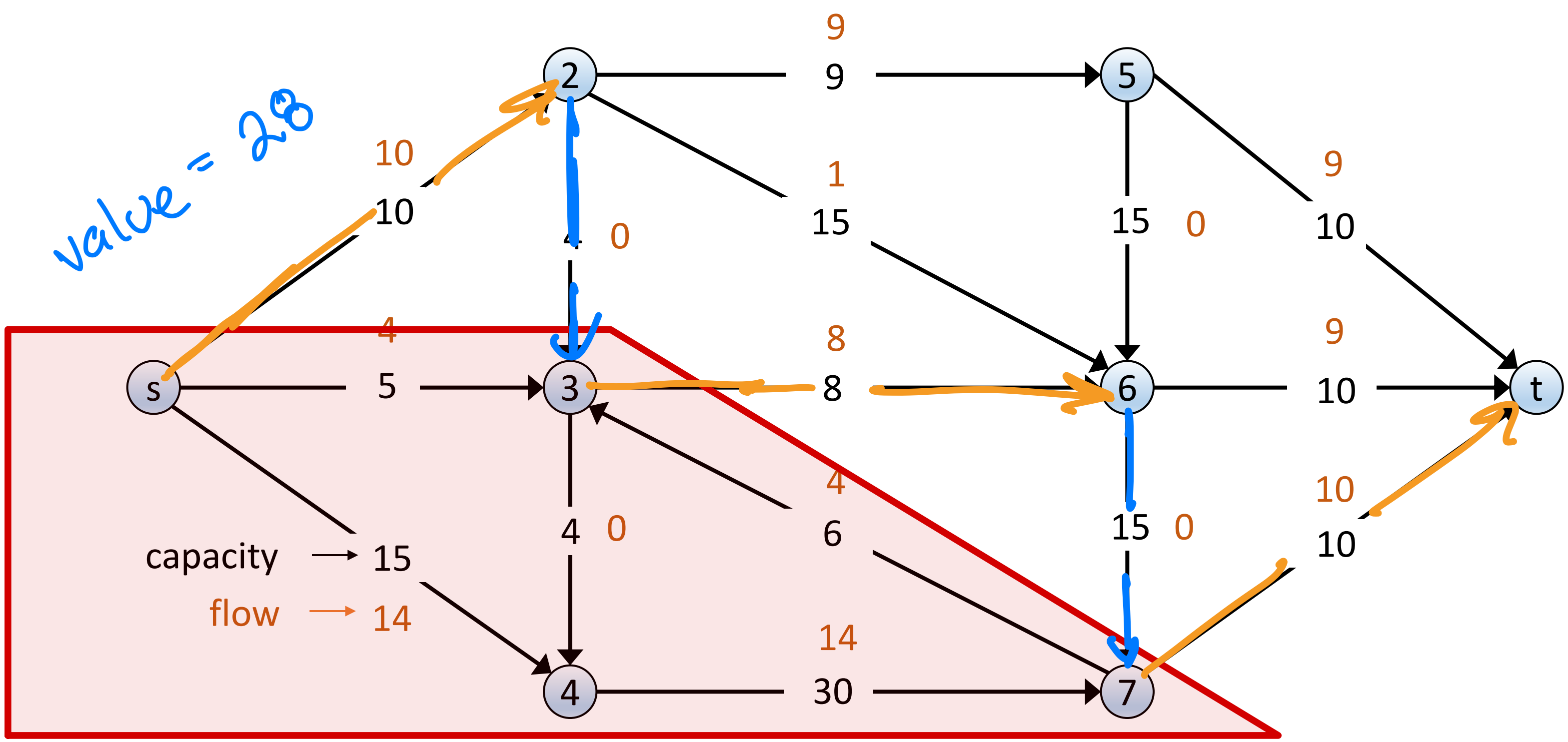
  except for s,t

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



# Maximum Flow Problem

Given $G = (V, E, s, t, \{c(e)\})$, find an s-t flow of max. value

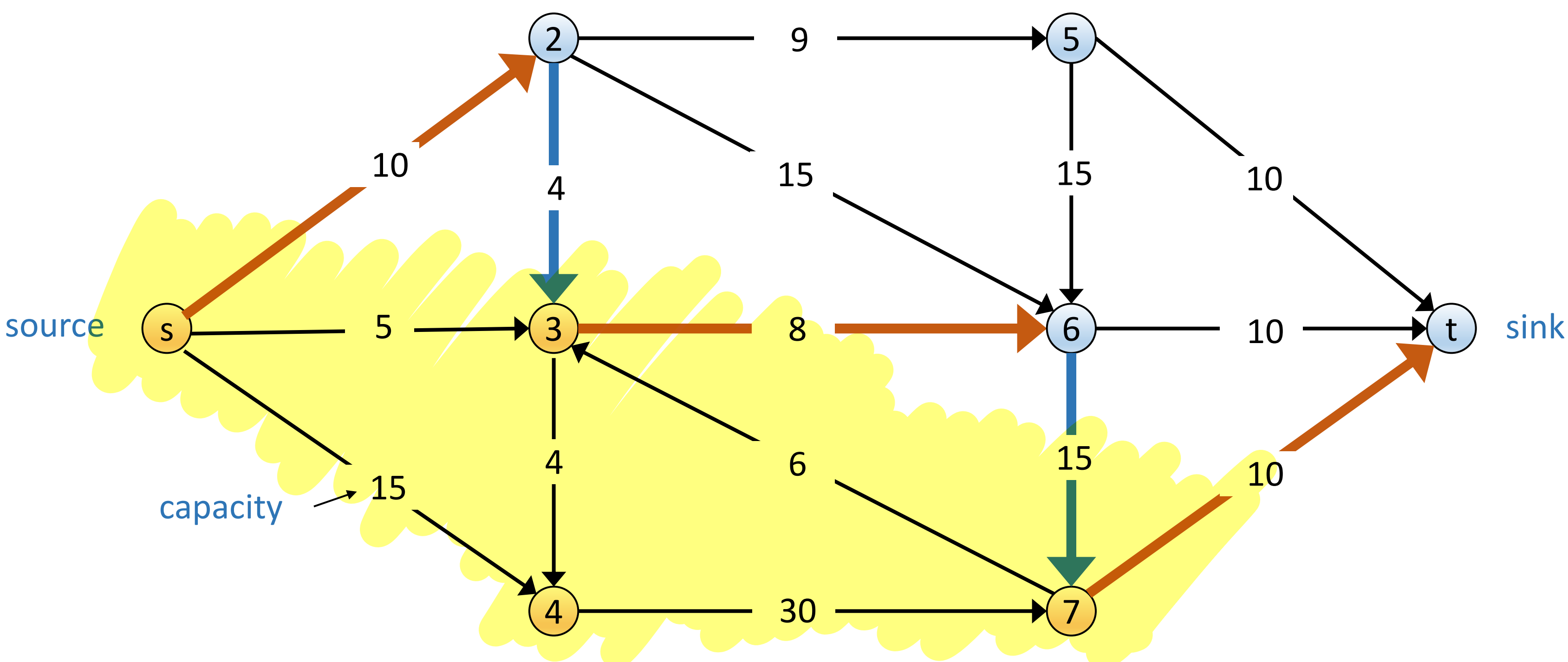


# Cuts

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

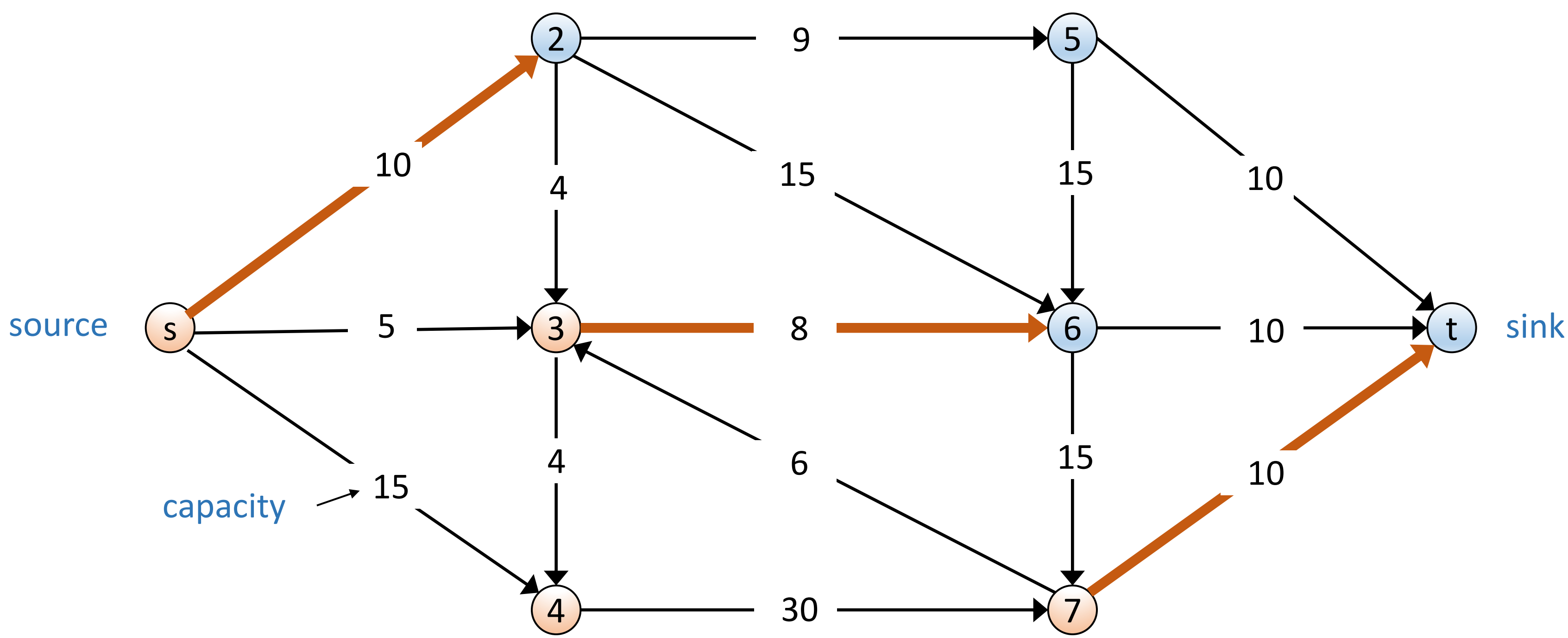
- The capacity of a cut $(A, B)$ is $cap(A, B) = \sum_{e \text{ out of } A} c(e)$

$cap(A,B) = 28$



# Minimum Cut problem

Given $G = (V, E, s, t, \{c(e)\})$, find an s-t cut of min. capacity



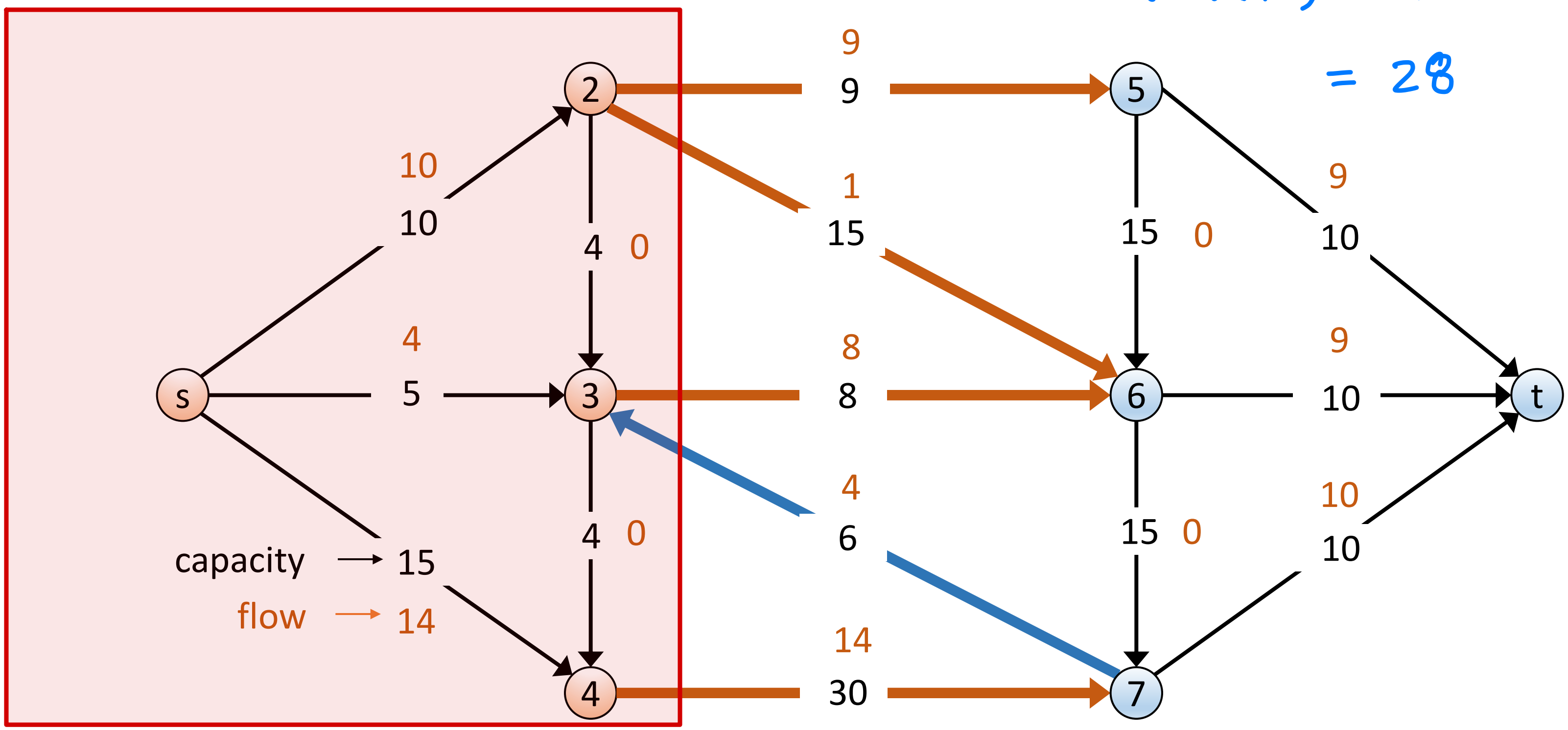
# Flows vs. Cuts

- **Fact:** If $f$ is any s-t flow and $(A, B)$ is any s-t cut, then the net flow across $(A, B)$ is equal to the amount leaving s

$$\sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) = val(f)$$

val(f) = 9 + 1 + 8 - 4 + 14

= 28



# Weak MaxFlow-MinCut Duality

- For any s-t flow $f$ and any s-t cut $(A, B)$ $val(f) \leq cap(A, B)$

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

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

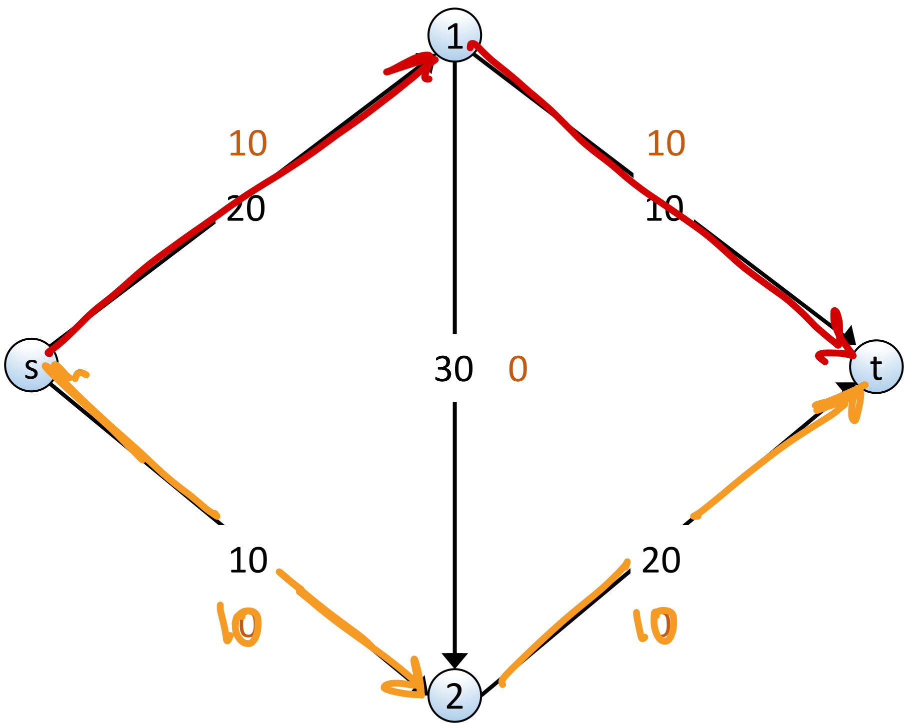
$$\leq \sum_{e \text{ out of } A} c(e) - \sum_{e \text{ into } A} f(e) \quad \text{(capacity)}$$

$$\leq \sum_{e \text{ out of } A} c(e) - 0 = cap(A,B) \quad \text{(non-negativity)}$$

- If $f$ is a flow, $(A, B)$ is a cut, and $val(f) = cap(A, B)$, then $f$ is a max flow and $(A, B)$ is a min cut

# Augmenting Paths

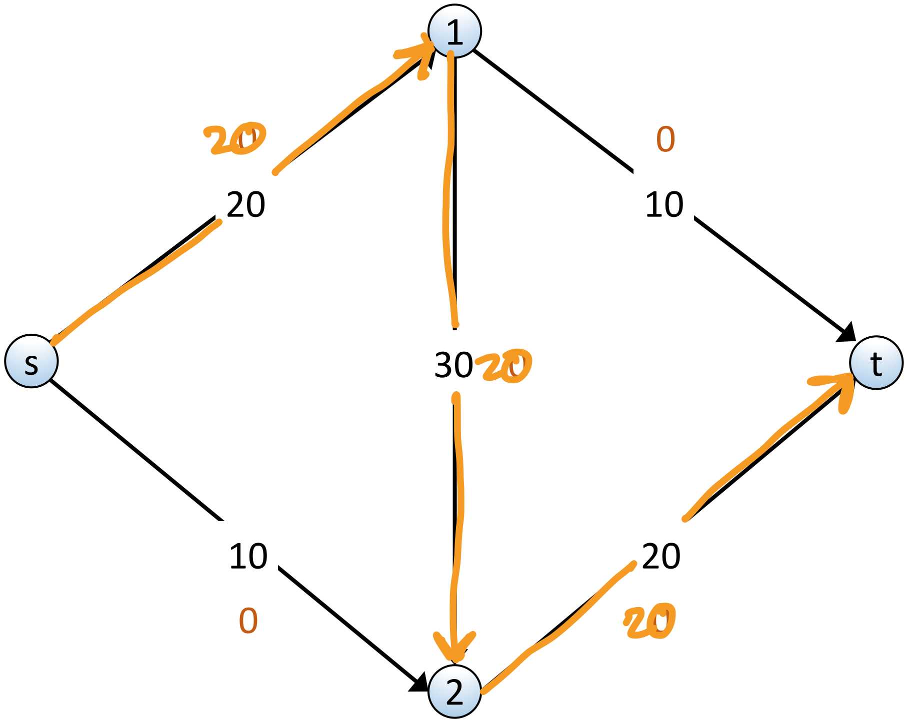
- Given a network $G = (V, E, s, t, \{c(e)\})$ and a flow $f$, an **augmenting path** $P$ is an $s \to t$ path such that $f(e) < c(e)$ for every edge $e \in P$



# Greedy Max Flow

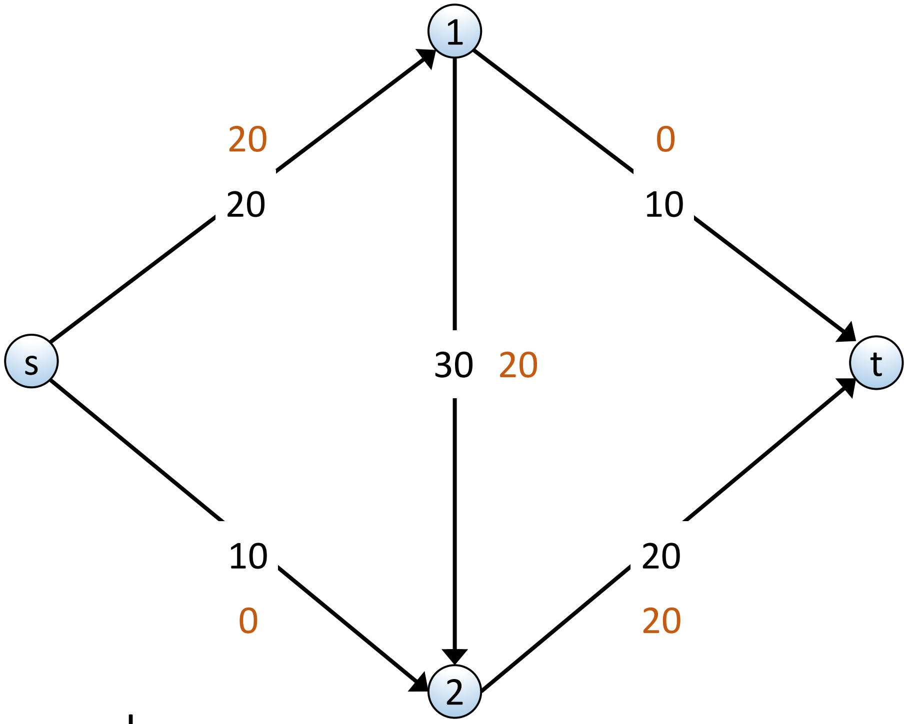
the only feasible flow you know

- Start with $f(e) = 0$ for all edges $e \in E$
- Find an **augmenting path** $P$
- Repeat until you get stuck

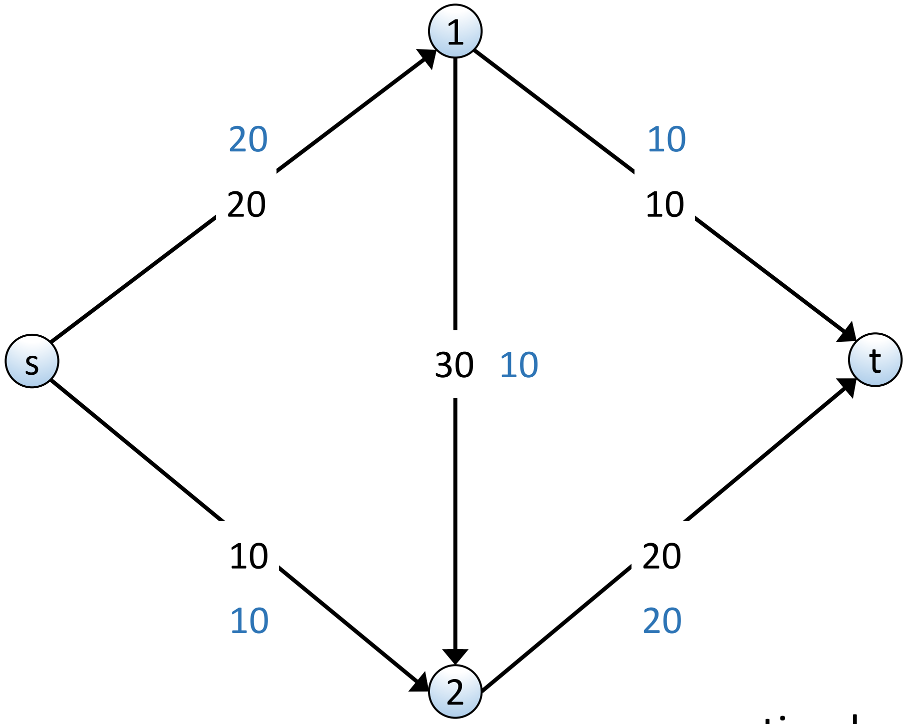


# Does Greedy Work?

- Greedy gets stuck before finding a max flow
- How can we get from our solution to the max flow?



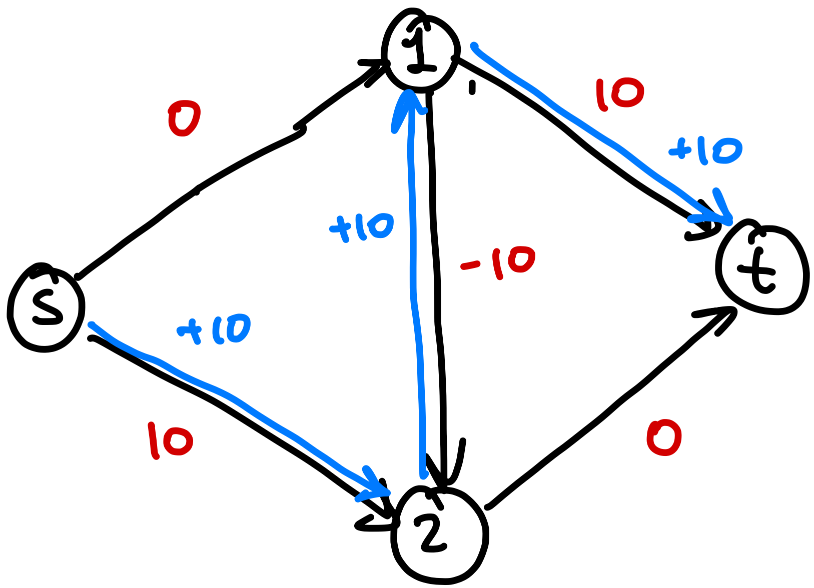

greedy



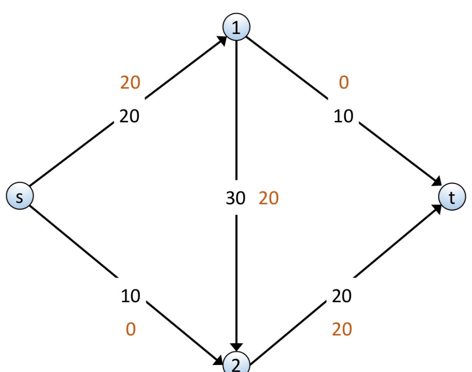

optimal

# Why didn't greedy work?

$f^* - f$

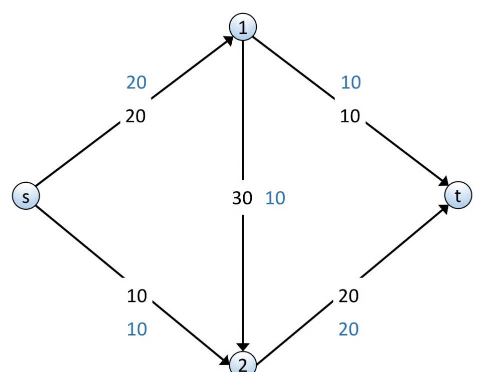



- Greedy send "negative flow"
- Negative flow $\approx$ sending flow back




greedy

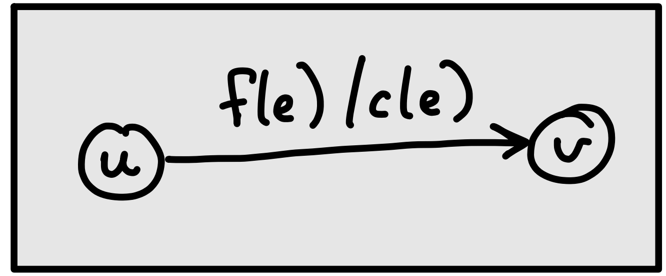
$f$



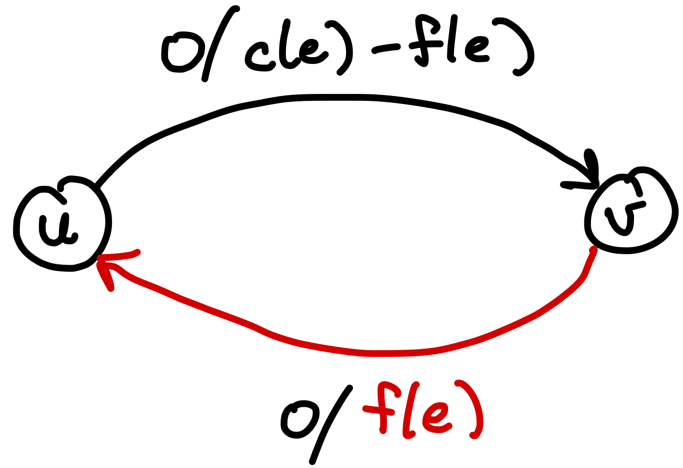

optimal

$f^*$

# Residual Graphs




- Original edge: $e = (u, v) \in E$.
  - Flow $f(e)$, capacity $c(e)$




- Residual edge
  - Allows "undoing" flow
  - $e = (u, v)$ and $e^R = (v, u)$.
  - Residual capacity

- Residual graph $G_f = (V, E_f)$
  - Edges with positive residual capacity.
  - $E_f = \{e : f(e) < c(e)\} \cup \{e^R : c(e) > 0\}$.

Given $G, f$ it's $O(m)$ time to compute $G_f$

# Augmenting Paths in Residual Graphs

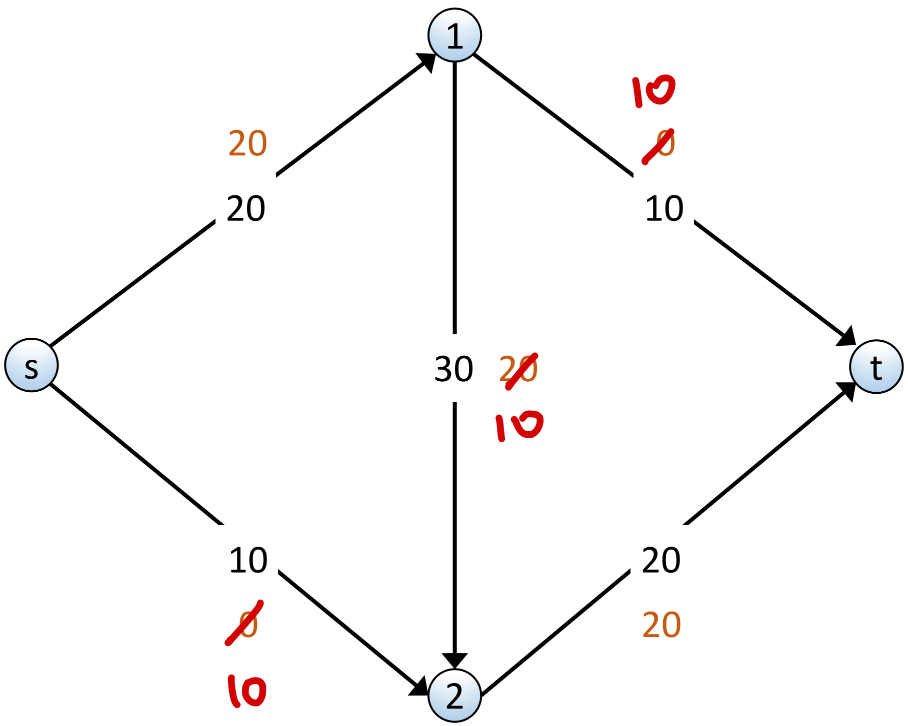
Any path where every edge has >0 capacity

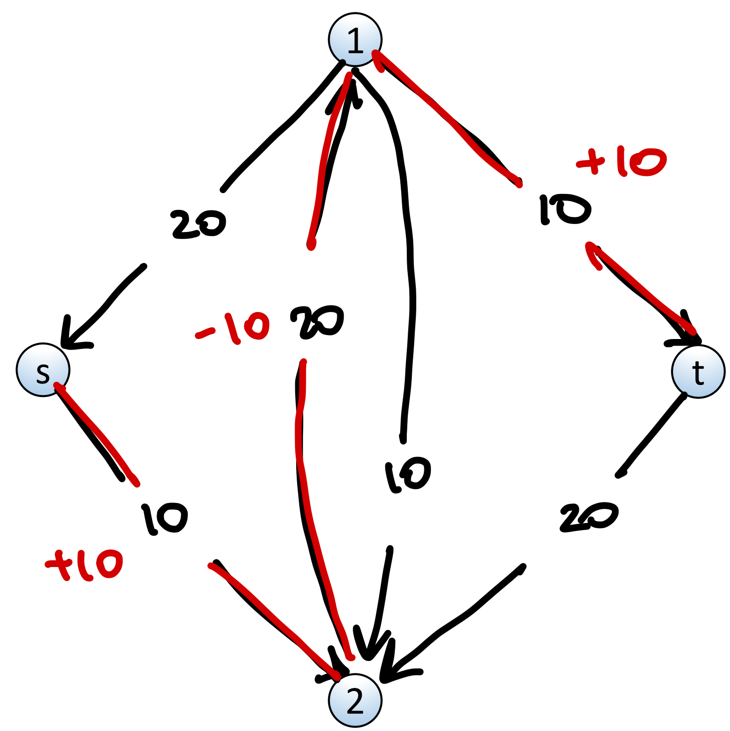
- Let $G_f$ be a **residual graph**
- Let $P$ be an augmenting path in the **residual graph**
- **Fact:** $f' = \text{Augment}(G_f, P)$ is a valid flow

```
Augment(Gf, P)
    b ← the minimum capacity of an edge in P
    for e ∈ P
        if e ∈ E:   f(e) ← f(e) + b
        else:       f(e) ← f(e) - b
    return f
```

# Ford-Fulkerson Algorithm

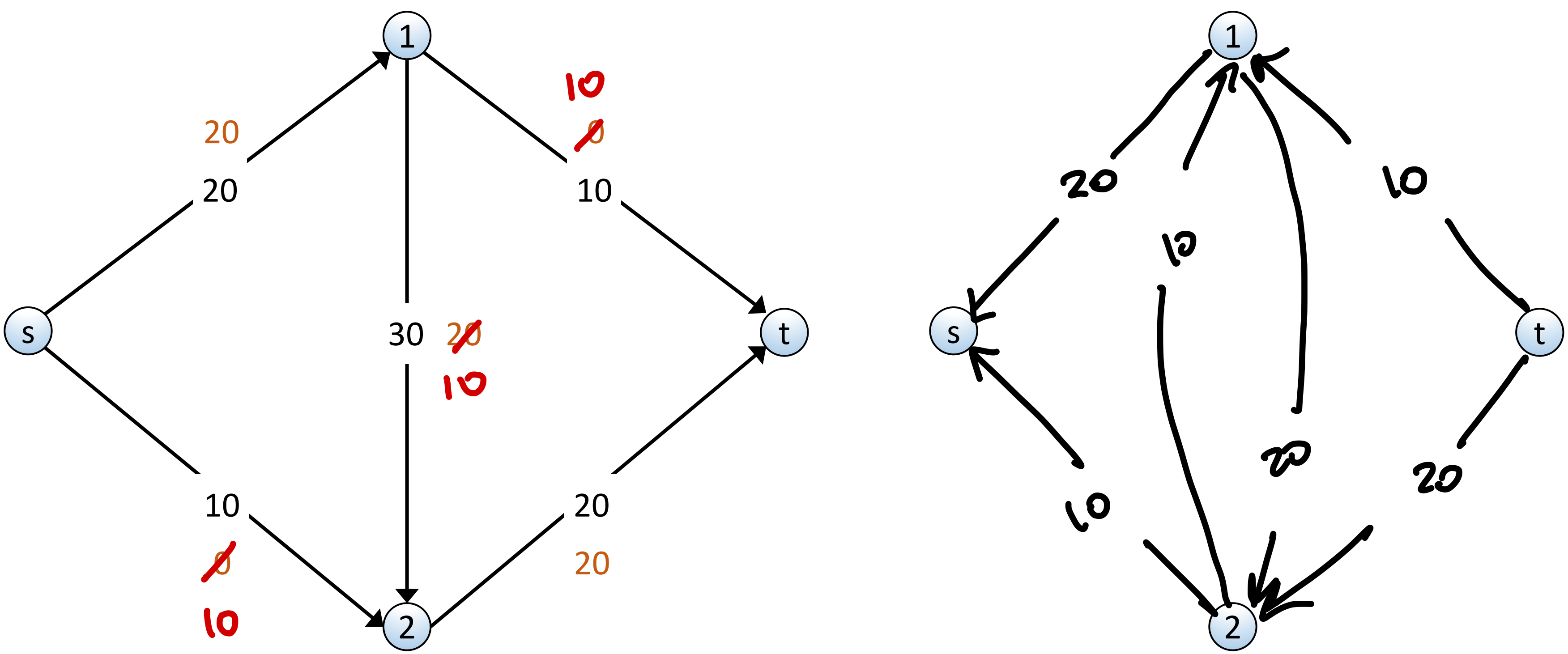
- Start with $f(e) = 0$ for all edges $e \in E$
- Find an **augmenting path** $P$ in the **residual graph**
- Repeat until you get stuck



# Ford-Fulkerson Algorithm

- Start with $f(e) = 0$ for all edges $e \in E$
- Find an **augmenting path** $P$ in the **residual graph**
- Repeat until you get stuck



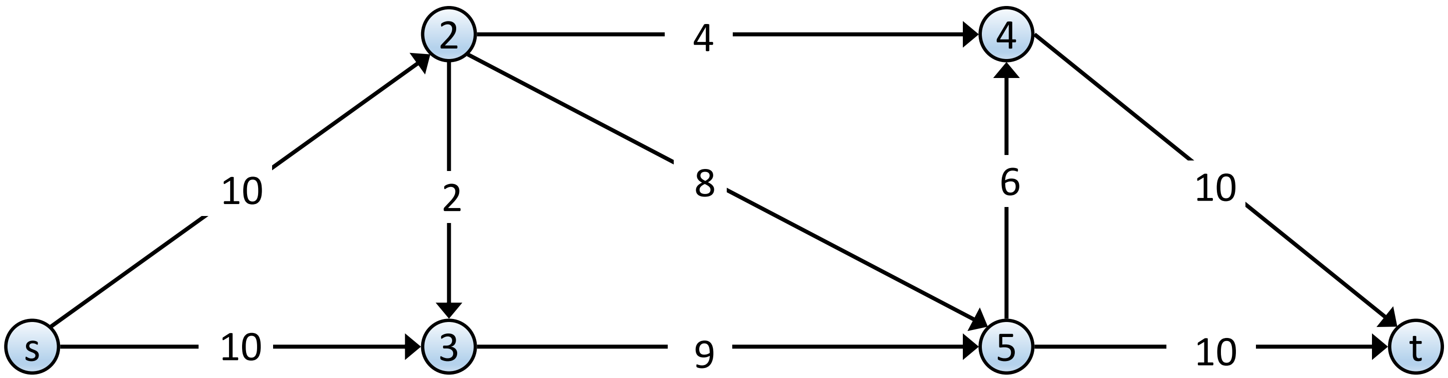
# Ford-Fulkerson Algorithm

```
FordFulkerson(G,s,t,{c(e)})
    for e ∈ E: f(e) ← 0              // O(m)
    Gf is the residual graph         // O(m)

    while (there is an s-t path P in Gf)
        f ← Augment(Gf,P) // O(m) per iteration      How many iterations?
        update Gf

    return f
```

```
Augment(Gf, P)
    b ← the minimum capacity of an edge in P
    for e ∈ P
        if e ∈ E:   f(e) ← f(e) + b
        else:       f(e) ← f(e) - b
    return f
```

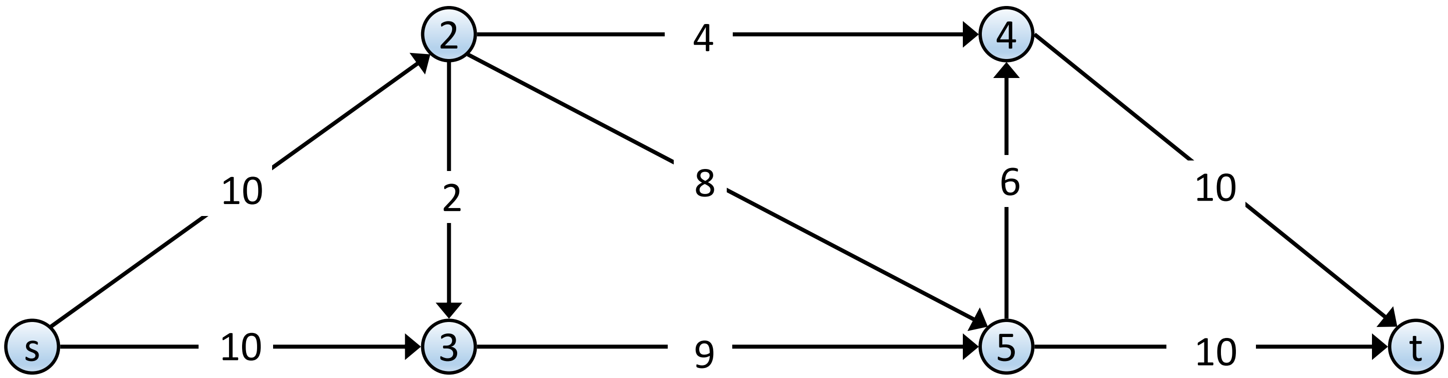
# Ford-Fulkerson Demo

$G$:



# Ford-Fulkerson Demo

$G$:




$G_f$:

# What do we want to prove?

- Termination?
- Correctness / optimality?
- Running time?
- Finding min cuts?

# Termination of Ford-Fulkerson

Assume all capacities are integers $0 \le c(e) \le C$

① Value of the max flow

$$val(f^*) \le nC$$

② Every augmentation adds $\ge 1$ to $val(f)$

$\Rightarrow$ # of iterations is $\le nC$

# Correctness of Ford-Fulkerson

- **Theorem:** the following are equivalent for all $f$
  1. There exists a cut $(A, B)$ such that $val(f) = cap(A, B)$
  2. Flow $f$ is a maximum flow
  3. There is no augmenting path in $G_f$

Strong max flow min cut duality

# Correctness of Ford-Fulkerson

- **Theorem:** $f$ is a maximum s-t flow if and only if there is no augmenting s-t path in $G_f$

- **Strong MaxFlow-MinCut Duality:** The value of the max s-t flow equals the capacity of the min s-t cut

- We'll prove that the following are equivalent for all $f$
  1. There exists a cut $(A, B)$ such that $val(f) = cap(A, B)$
  2. Flow $f$ is a maximum flow
  3. There is no augmenting path in $G_f$

# Correctness of Ford-Fulkerson

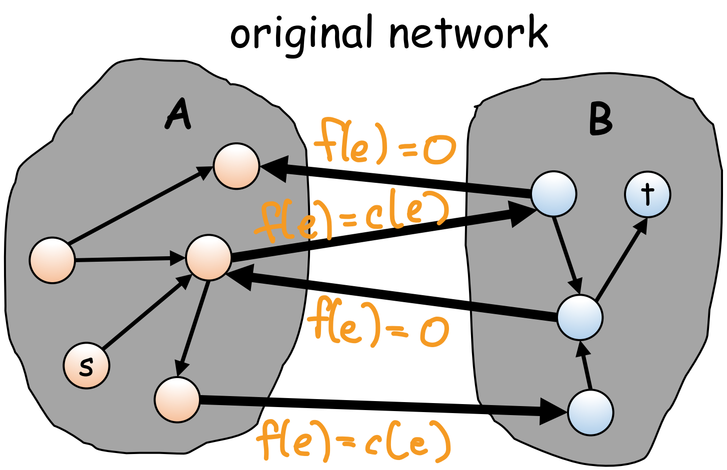
- **(3 → 1)** If there is no augmenting path in $G_f$, then there is a cut $(A, B)$ such that $val(f) = cap(A, B)$
  - Let $A$ be the set of nodes reachable from $s$ in $G_f$
  - Let $B$ be all other nodes

notice $s \in A$ , $t \in B$

# Correctness of Ford-Fulkerson

- **(3 → 1)** If there is no augmenting path in $G_f$, then there is a cut $(A, B)$ such that $val(f) = cap(A, B)$
  - Let $A$ be the set of nodes reachable from $s$ in $G_f$
  - Let $B$ be all other nodes
  - **Key observation:** no edges in $G_f$ go from $A$ to $B$

- If $e$ is $A \to B$, then $f(e) = c(e)$
- If $e$ is $B \to A$, then $f(e) = 0$




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

$$= \sum_{e \text{ out of } A} c(e) - \sum_{e \text{ into } A} 0$$

$$= cap(A, B)$$

# Running Time of Ford-Fulkerson

New Problem: Integer max flow

Given $G = (V, E, s, t, \{c(e)\})$ with integer $c(e)$

find a flow with integer $f(e)$

FF solves integer max flow

Running Time (for Integer Max Flow)

- $O(m \cdot val(f^*)) = O(mnC)$ largest capacity

- $O(mn)$ if all capacities are 1

# Summary