HW7 due tonight
HW8 out tonight

MT II Tuesday Mar 27th
- graph algorithms

CS4800: Algorithms & Data
Jonathan Ullman

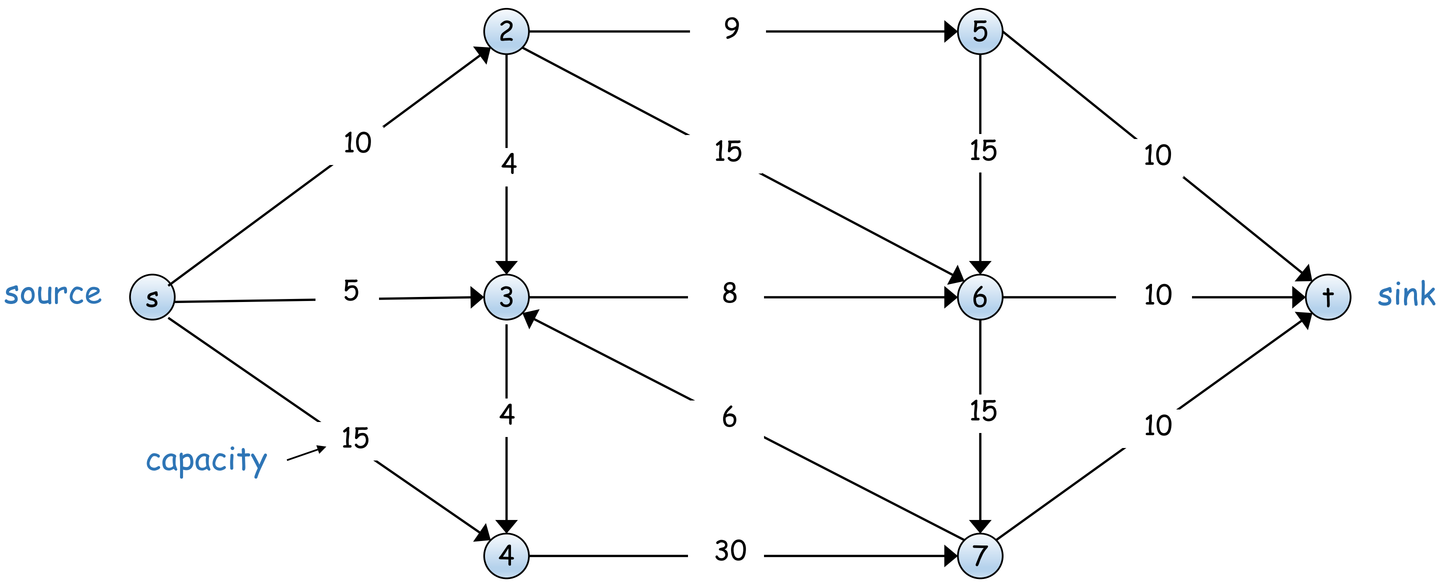
Lecture 16:
- Network Flow
  - MaxFlow-MinCut Duality
  - Ford Fulkerson

Mar ~~13~~, 2018
16

Flow Networks

Flow Networks

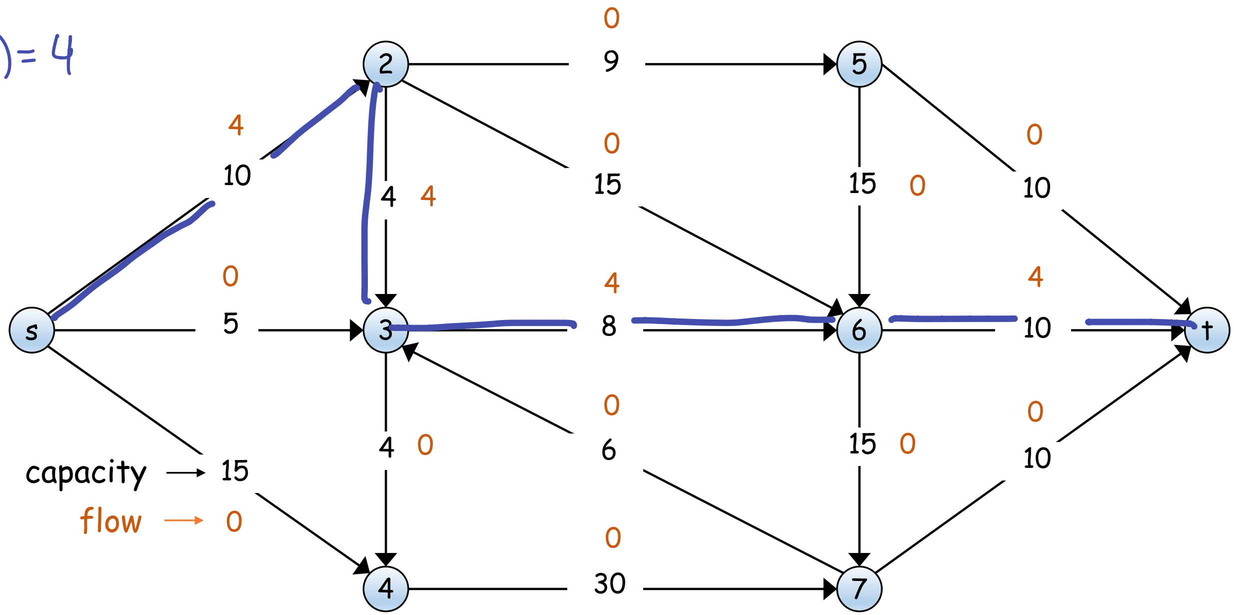
- Directed graph $G = (V, E)$
- Two special nodes: source s and sink = t
- Edge capacities $c(e)$



Flows

- An s-t flow is a function $f(e)$ such that
  - For every $e \in E$, $0 \le f(e) \le c(e)$ (capacity) ✓
  - For every $\cancel{v \in E}$, $\sum_{e \text{ in to } v} f(e) = \sum_{e \text{ out of } v} f(e)$ (conservation)
    $v \in E \setminus \{s,t\}$
- The value of a flow is $val(f) = \sum_{e \text{ out of } s} f(e)$

$val(f) = 4$

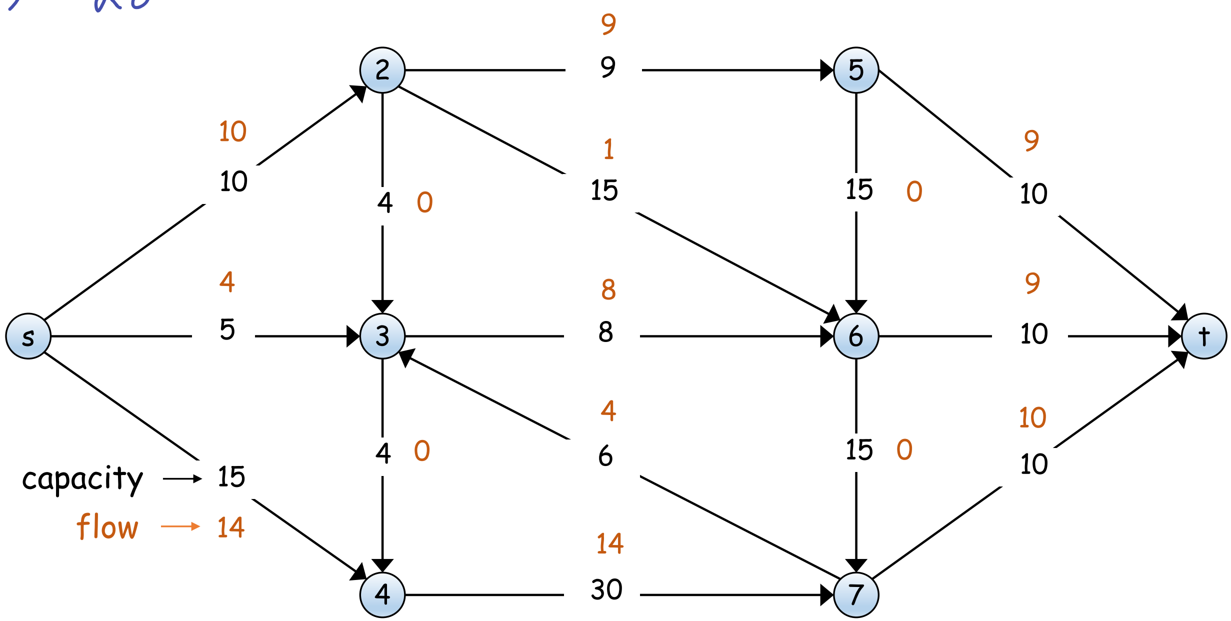


Maximum Flow Problem

- Find an s-t flow of maximum value

maximum over f satisfying capacity / conservation
of val(f)

val(f) = 28



Ask the Audience

- True or False? There is always a flow such that every edge e leaving the source s is saturated with $f(e) = c(e)$
  - Explain why or give a counterexample

$$s \xrightarrow{2} 2 \xrightarrow{1} 3 \xrightarrow{2} t$$

"obviously" the max flow is value 1.

Cuts

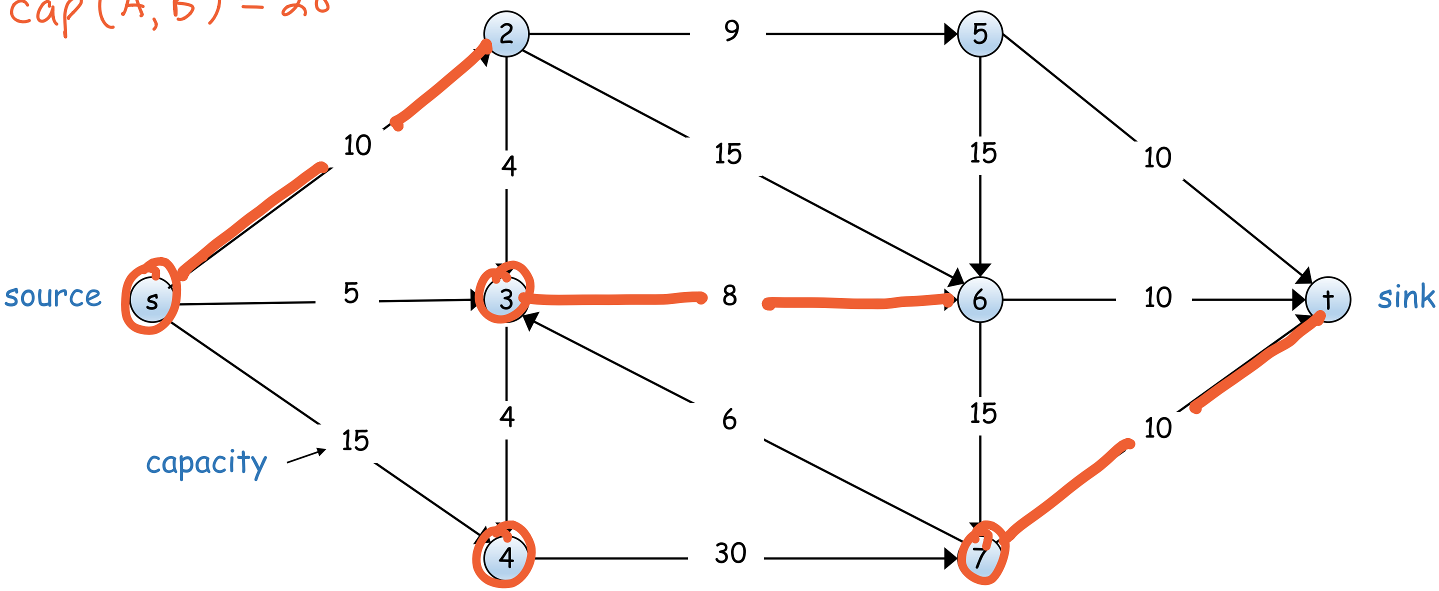
- An s-t cut is a partition (A, B) of V with $s \in A$ and $t \in B$

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

$\{e = (u,v) : u \in A,\ v \in B\}$

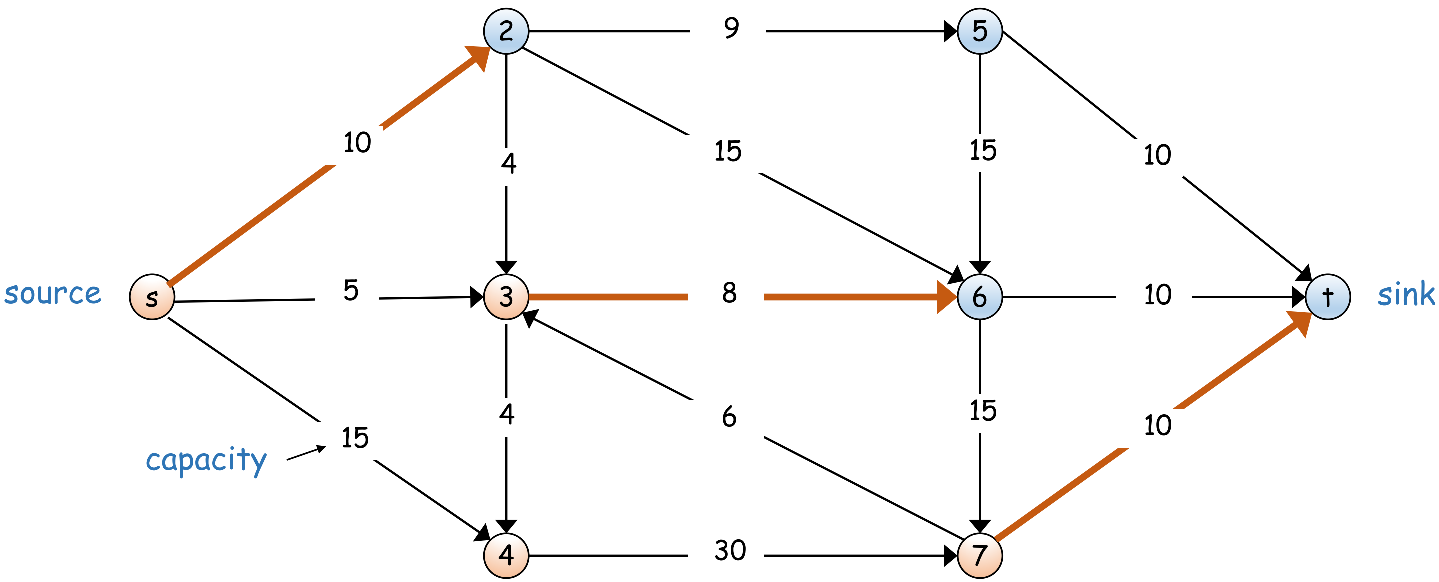
$A = \{s, 3, 4, 7\}$

$cap(A, B) = 28$



Minimum Cut problem

- Find an s-t cut of minimum 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

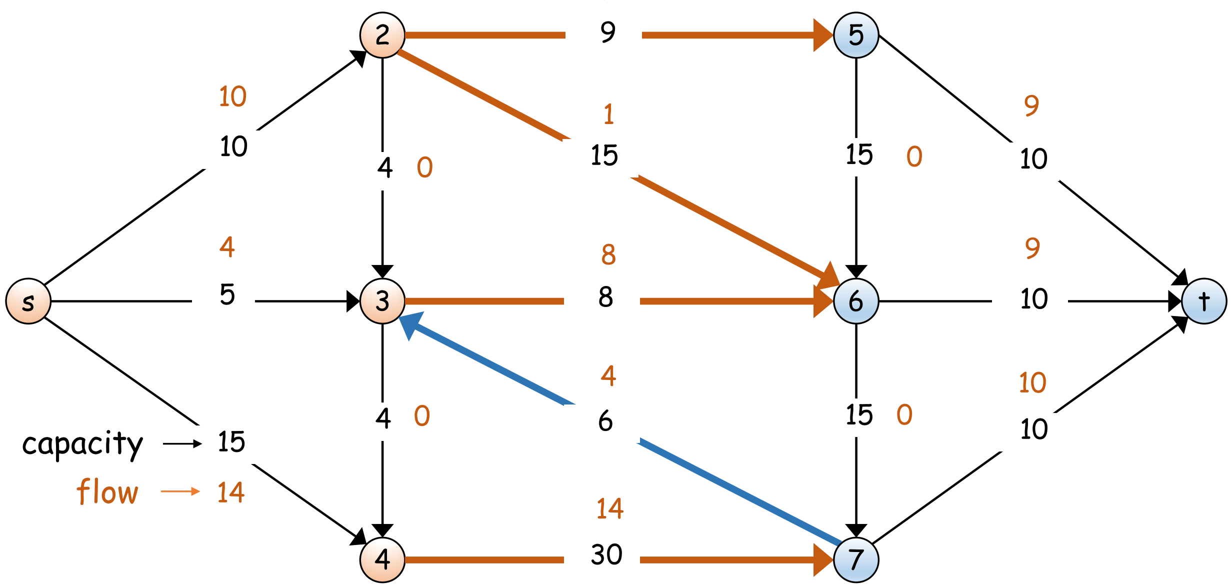
$$\sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) = val(f)$$

A = {s, 2, 3, 4}

B = {5, 6, 7, t}

val(28)

9+1+8+14−4=28



Flows vs. Cuts

max flow $\leq$ min cut

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

$$val(f) \leq cap(A, B)$$

$$\begin{aligned}
val(f) &= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e) && \text{(by Fact)} \\
&\leq \sum_{e \text{ out of } A} f(e) && \text{(by non-negativity)} \\
&\leq \sum_{e \text{ out of } A} c(e) && \text{(by cap constraint)} \\
&= cap(A, B) && \text{(by definition)}
\end{aligned}$$

Flows vs. Cuts

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

$$val(f) \leq cap(A, B)$$

Observation: If f is a flow and (A,B) is a cut and val(f) = cap(A,B) then f is a max flow and (A,B) is a min cut.

Greedy Max Flow

a feasible flow

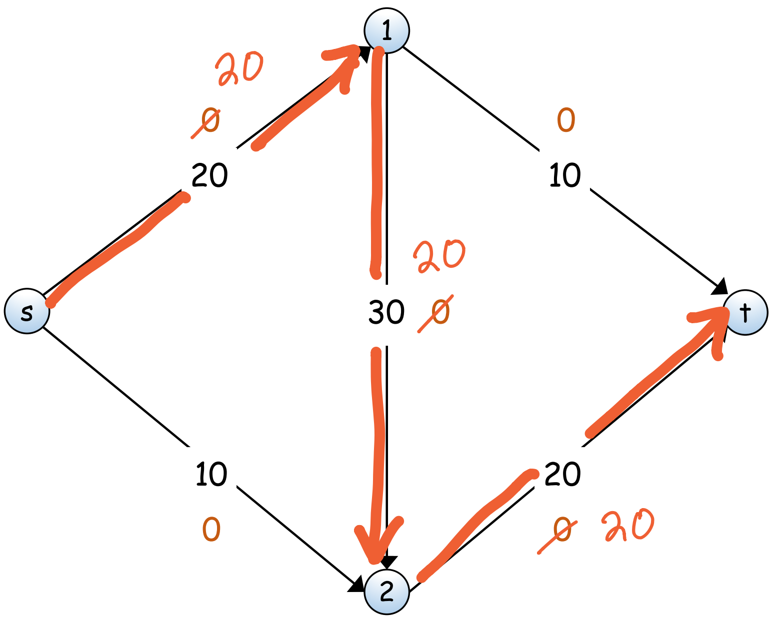
- Start with $f(e) = 0$ for all edges $e \in E$
- Find an s-t path P where every edge has $f(e) < c(e)$
- Augment flow along the path P
- Repeat until you get stuck

- find the "bottleneck"

$$b = \min_{e \in P} c(e) - f(e)$$

- add b to each edge

each path is an augmenting path and the process is called an augmentation



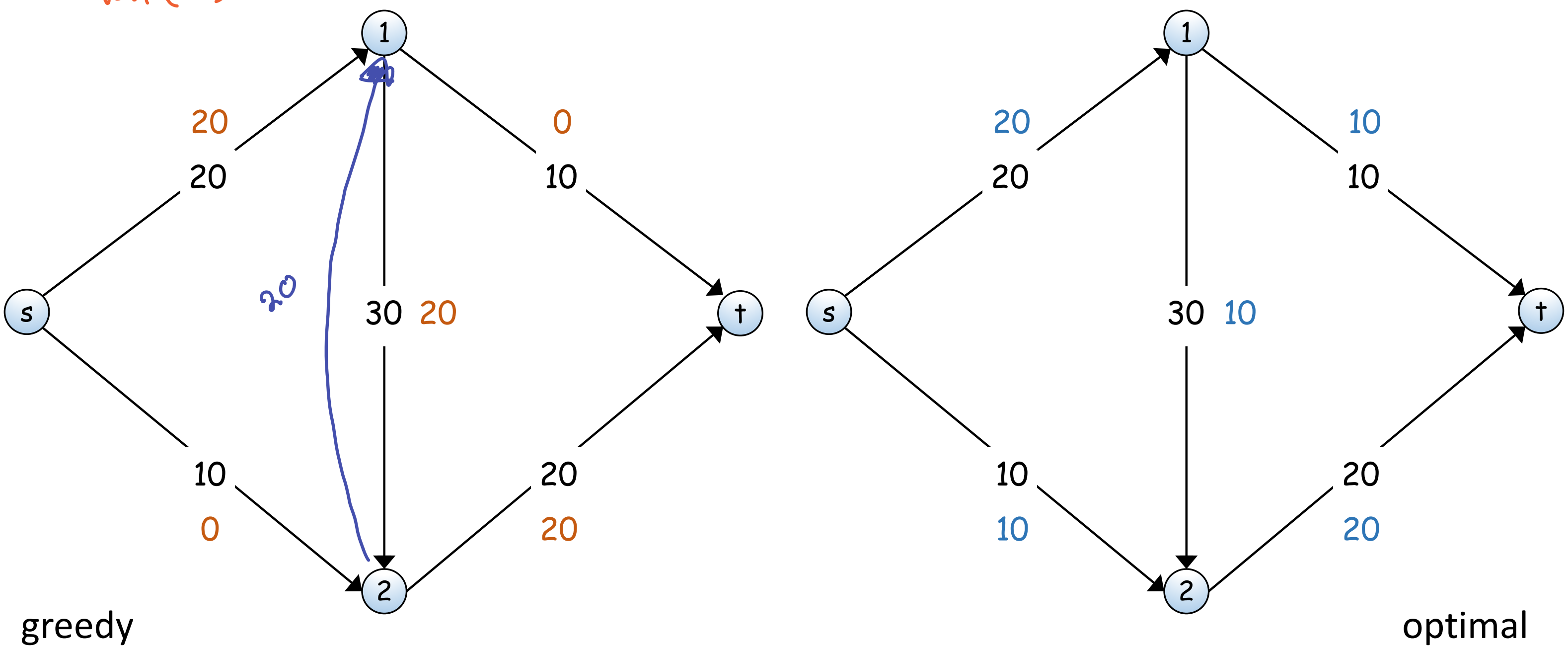
Does Greedy Work?

- Take I gets stuck before finding the optimum
- How can we get from our solution to the optimum?

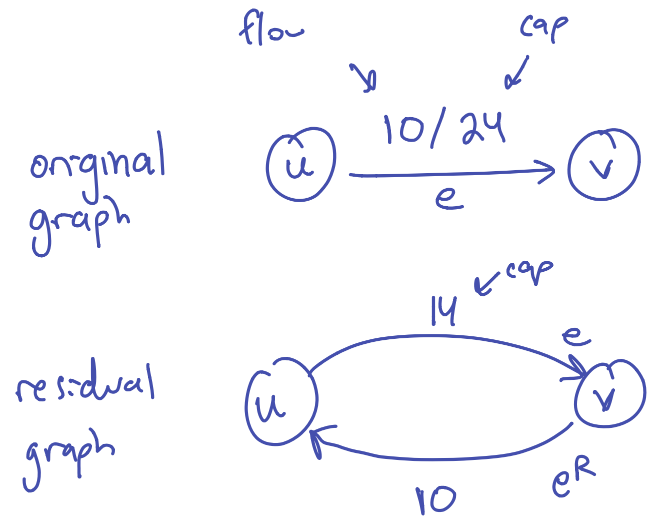
no augmenting paths

val(f) = 20

val(f) = 30



Residual Graphs



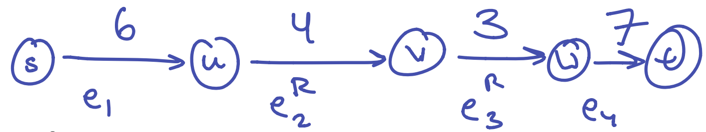
- Original edge: $e = (u, v) \in E$.
  - Flow $f(e)$, capacity $c(e)$

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

depends on the flow

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

Augmenting Paths in Residual Graphs



- Let G_f be a residual graph
- Let P be a 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
```

$b = 3$

$f(e_1) \mathrel{+}= 3$

$f(e_2) \mathrel{-}= 3$

$f(e_3) \mathrel{-}= 3$

$f(e_4) \mathrel{+}= 3$

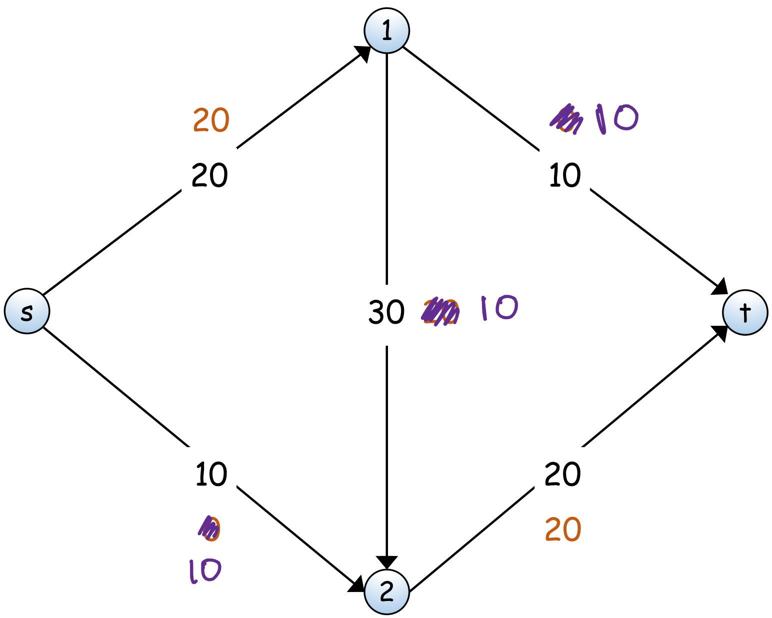
Key Fact: If f is feasible then $\text{Augment}(G_f, P)$ is also feasible.

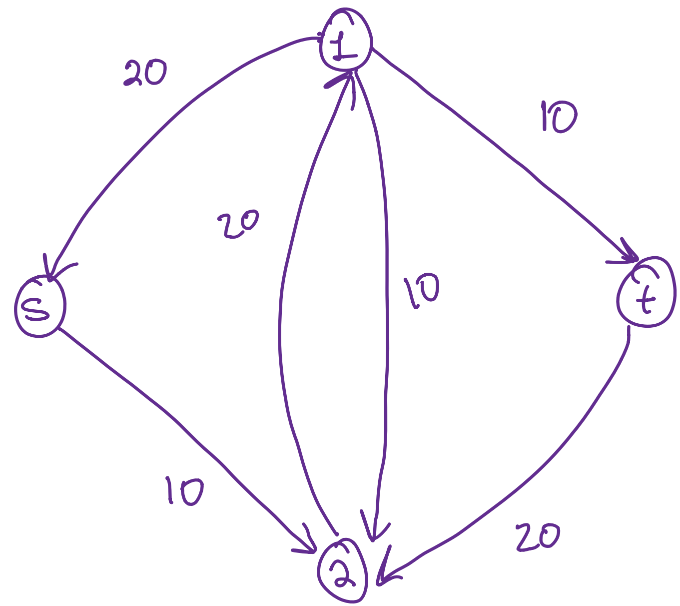
Ford-Fulkerson Algorithm

Note: conservation still satisfied

- Start with $f(e) = 0$ for all edges $e \in E$
- Find an s-t path P in the residual graph G_f
- Augment flow along the path P
- Repeat until you get stuck

$$s \xrightarrow{10} 2 \xrightarrow{20} 1 \xrightarrow{10} t$$





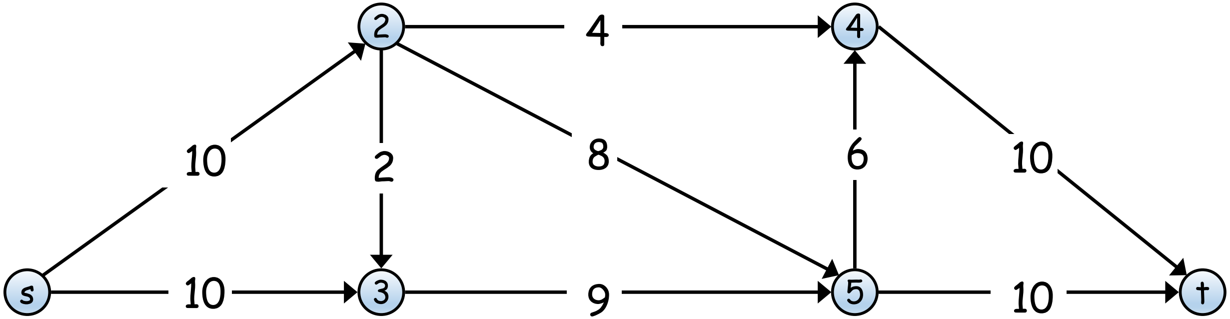
Ford-Fulkerson Algorithm

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

    while (there is an s-t path P in G_f)
        f ← Augment(G_f,P)
        update G_f

    return f
```

```
Augment(G_f, 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:



G_f:

What do we want to prove?

- Feasibility: FF outputs a feasible flow ✓
- Maximality / Terminates:
- Running Time:

Running Time of Ford-Fulkerson

Assumption: G has integer capacities (termination is clear)

- For integer capacities, $\leq val(f^*)$ augmentation steps

- Can perform each augmentation step in $O(m)$ time
  - find augmenting path in $O(m)$ (BFS)
  - augment the flow along path in $O(n)$
  - update the residual graph along the path in $O(n)$

- For integer capacities, FF runs in $O\big(m \cdot val(f^*)\big)$ time
  - $O(mn)$ time if all capacities are $c_e = 1$
  - $O(mnC_{\max})$ time for any integer capacities
  - Problematic when capacities are large—more on this later!

On tuesday we'll do better using careful augmenting paths

Optimality 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 If FF terminates, f is max flow

- MaxFlow-MinCut Duality: The value of the maximum s-t flow equals the capacity of the minimum 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

Optimality 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

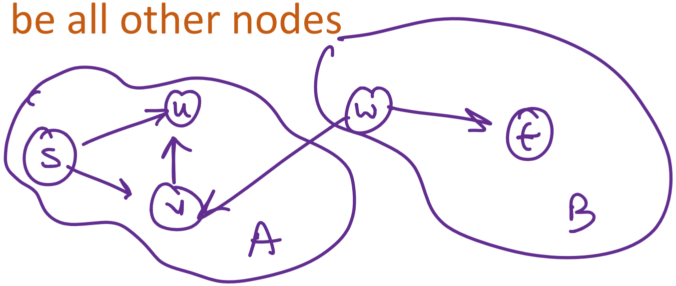
(1) $\Rightarrow$ (2) $\forall f, (A,B)$ $val(f) \leq cap(A,B)$

(2) $\Rightarrow$ (3) If there were an augmenting path in G_f then Augment(G_f, P) is better, so f is not max

hard part is (3) $\Rightarrow$ (1)

Optimality 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



Observation:

In G_f, there are no edges from A to B.

Optimality 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 (but in G some can)

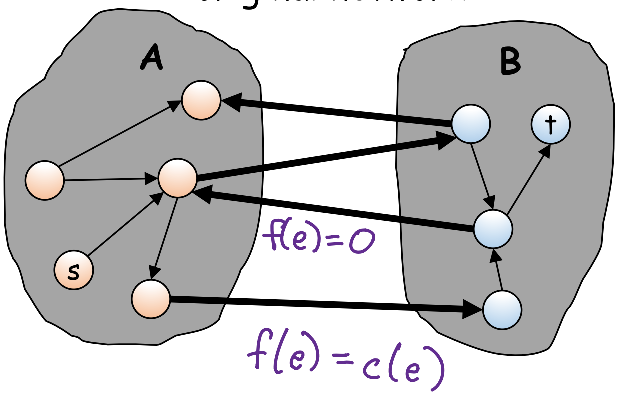
- If e is $A \to B$, then $f(e) = c(e)$
- If e is $B \to A$, then $f(e) = 0$

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

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

$$= cap(A, B)$$

original network



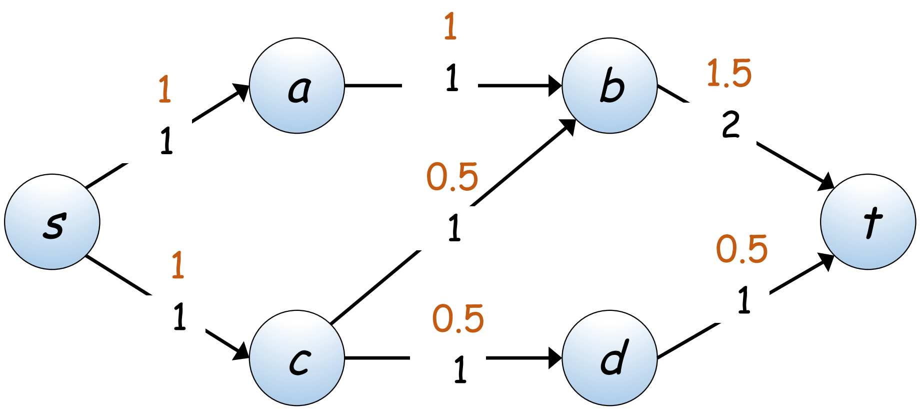
Summary

- The Ford-Fulkerson Algorithm solves maximum s-t flow
  - Running time $O\left(m \cdot val(f^*)\right)$ in networks with integer capacities
  - Space $O(n+m)$

- MaxFlow-MinCut Duality: The value of the maximum s-t flow equals the capacity of the minimum s-t cut
  - If f^* is a maximum s-t flow, then the set of nodes reachable from s in G_{f^*} gives a minimum cut
  - Given a max-flow, can find a min-cut in time $O(n+m)$

Ask the Audience

- Is this a maximum flow?



- Is there an **integral maximum flow**?
- Does every graph with integral capacities have an integral maximum flow?

Summary

- The Ford-Fulkerson Algorithm solves maximum s-t flow
  - Running time $O\big(m \cdot val(f^*)\big)$ in networks with integer capacities
  - Space $O(n+m)$
- MaxFlow-MinCut Duality: The value of the maximum s-t flow equals the capacity of the minimum s-t cut
  - If f^* is a maximum s-t flow, then the set of nodes reachable from s in G_{f^*} gives a minimum cut
  - Given a max-flow, can find a min-cut in time $O(n+m)$
- Every graph with integral capacities has an integral maximum flow
  - Ford-Fulkerson will return an integral maximum flow