CS3000: Algorithms & Data
Jonathan Ullman

Lecture 17:

- Network Flow: flows, cuts, duality
- Ford-Fulkerson

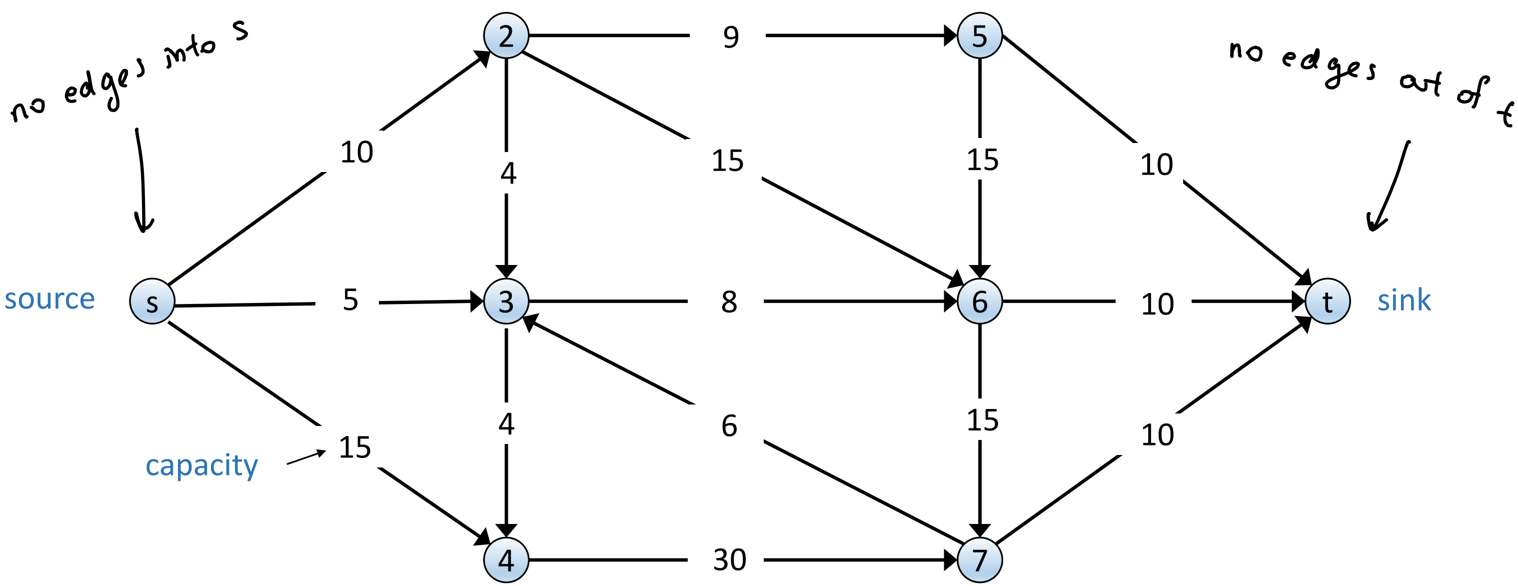
Nov 6, 2018

Flow Networks

Flow Networks

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

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



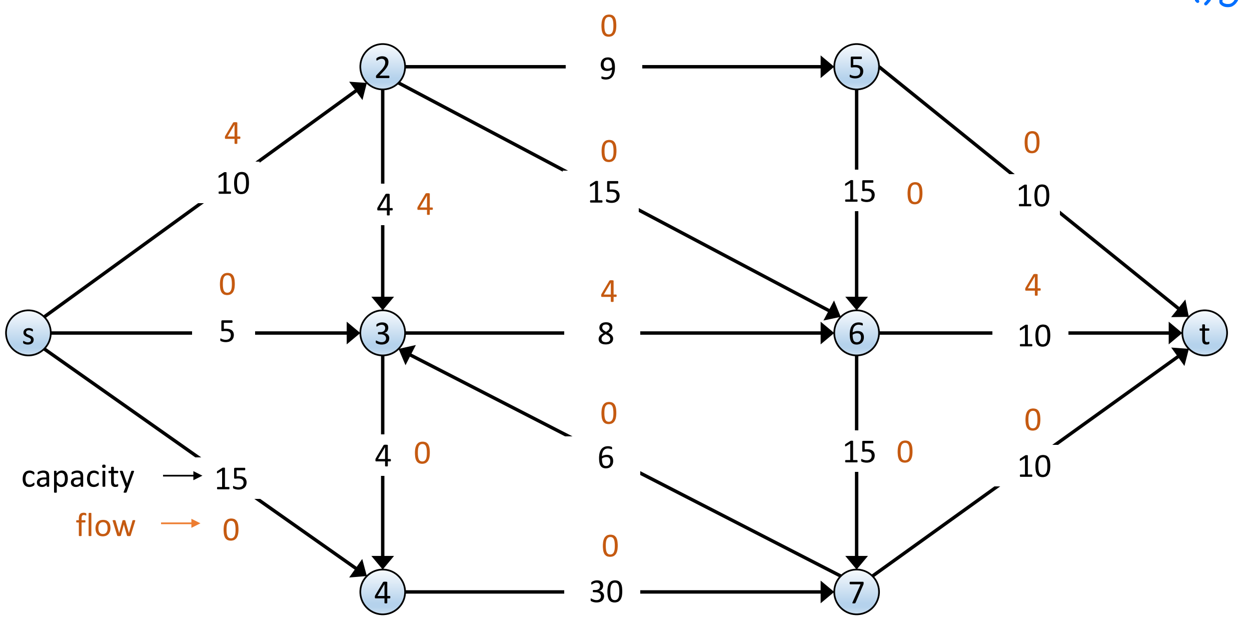
Flows

non-negativity

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

    $v \in V$, $v \neq s,t$
- The value of a flow is $val(f) = \sum_{e \text{ out of } s} f(e)$

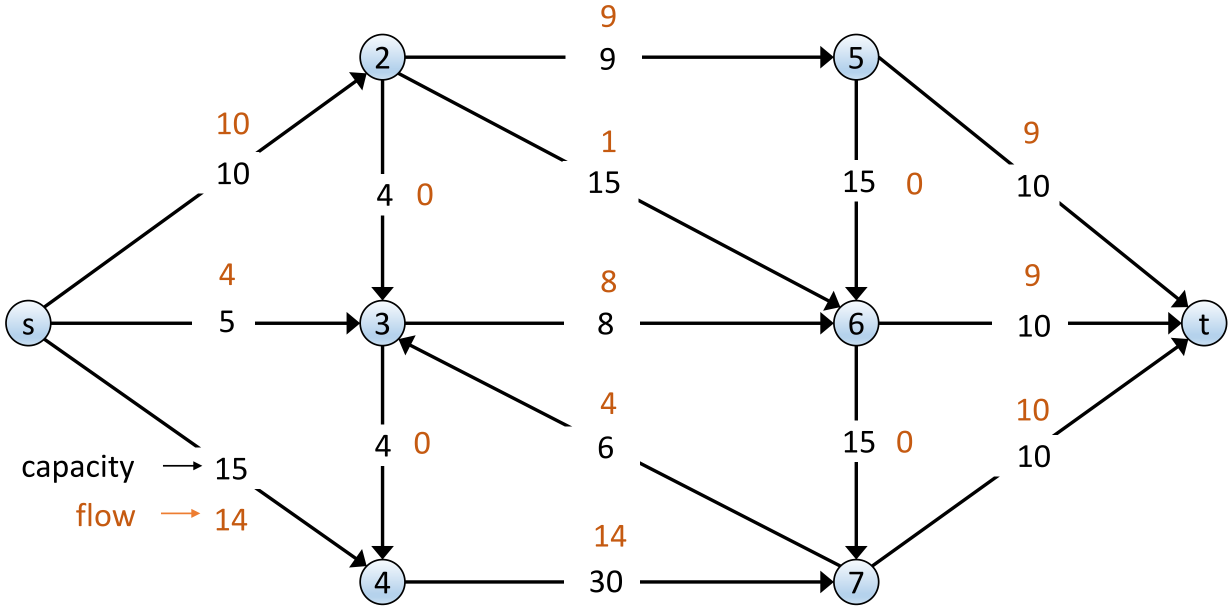
  $val : \mathbb{R}^m_{\geq 0} \to \mathbb{R}_{\geq 0}$



Maximum Flow Problem

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

$val(f) = 10 + 4 + 14 = 28$

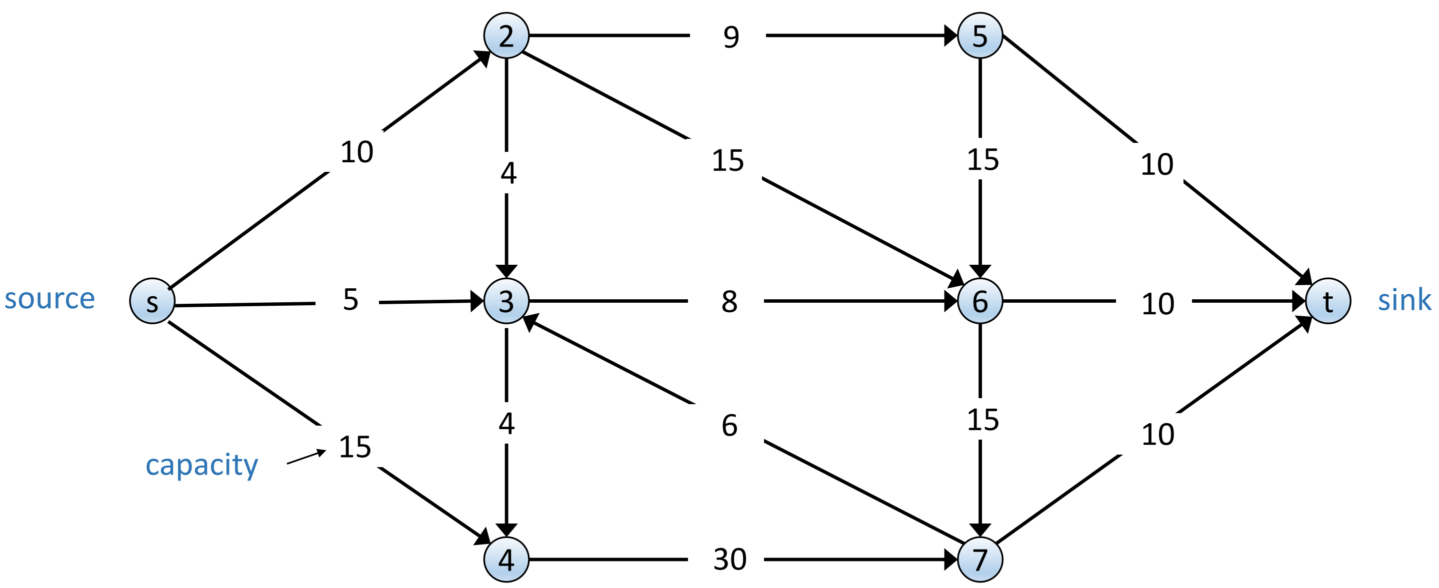


Cuts

$A \cap B = \emptyset$
$A \cup B = V$

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



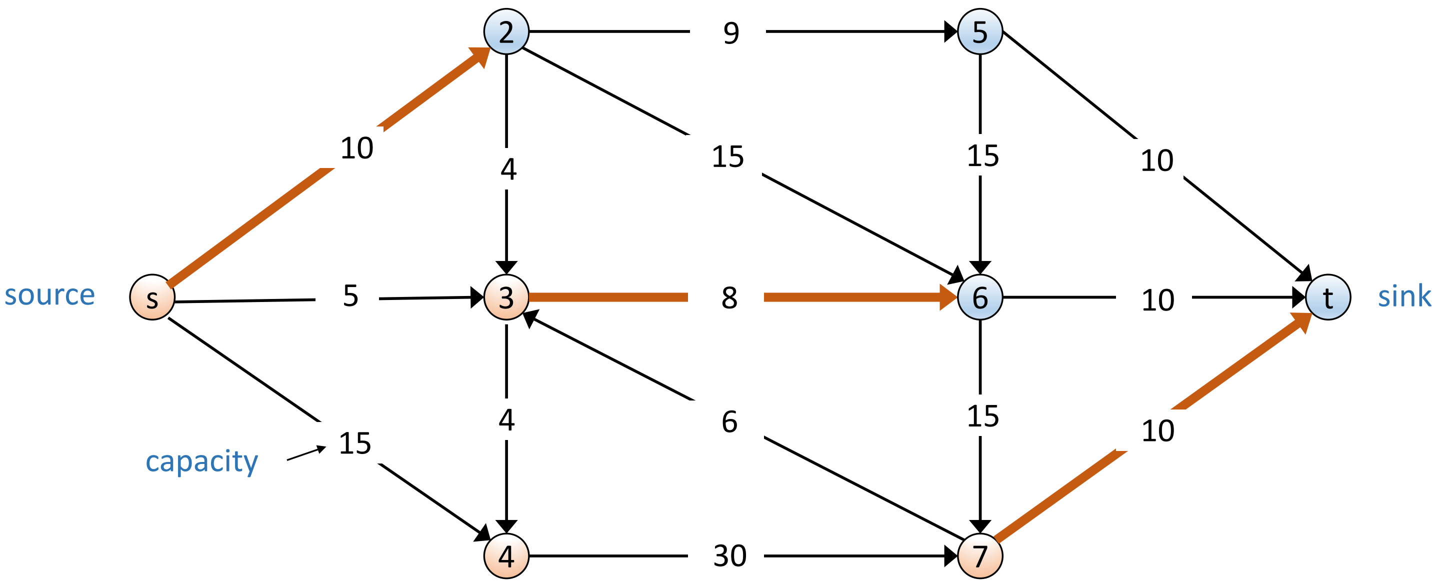
Minimum Cut problem

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

A = orange
B = blue

orange edges are "out of A"

cap(A, B) = 10 + 8 + 10 = 28

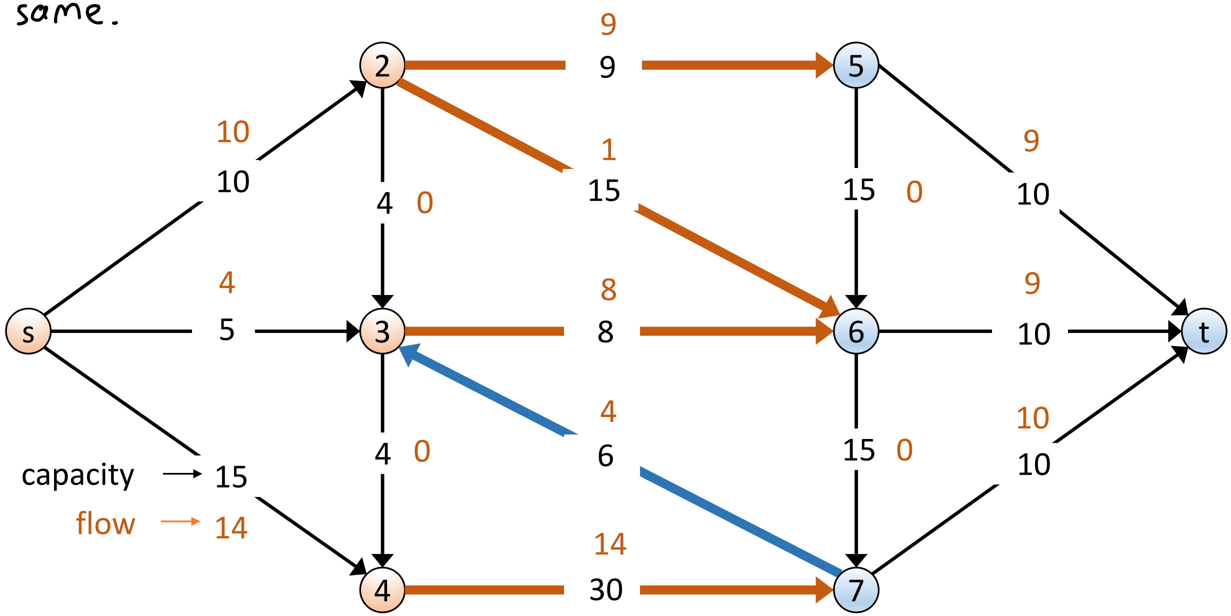


Flows vs. Cuts

- **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

The net flow across any s-t cut is the same.

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



Flows vs. Cuts

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

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

Proof:

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

$$\le \sum_{e \text{ out of } A} f(e)$$ [Non-negativity]

$$\le \sum_{e \text{ out of } A} c(e)$$ [Capacity]

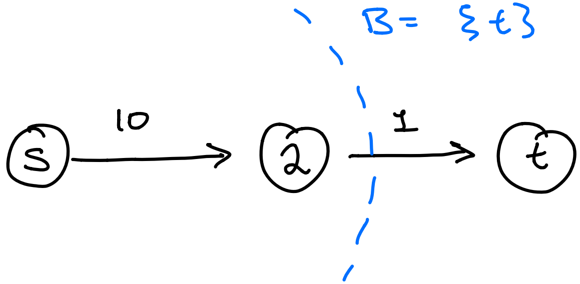
$$= cap(A, B)$$ [Definition]

Ask the Audience

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

$A = \{s, 2\}$ $\quad$ $cap(A, B) = 1$

$B = \{t\}$

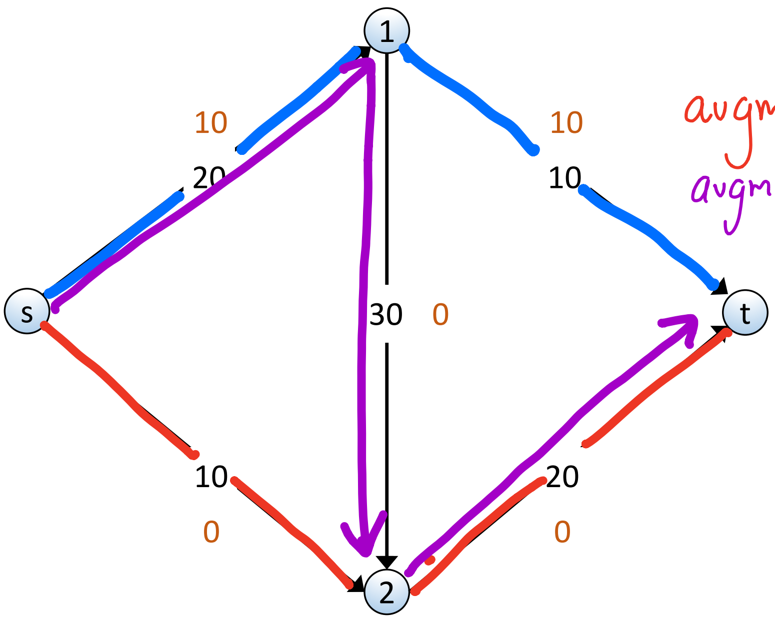


$$\sum_{e \text{ out of } \{s\}} c(e) = 10$$

$$val(f^*) = 1$$

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 $e \in P$



not an augmenting path

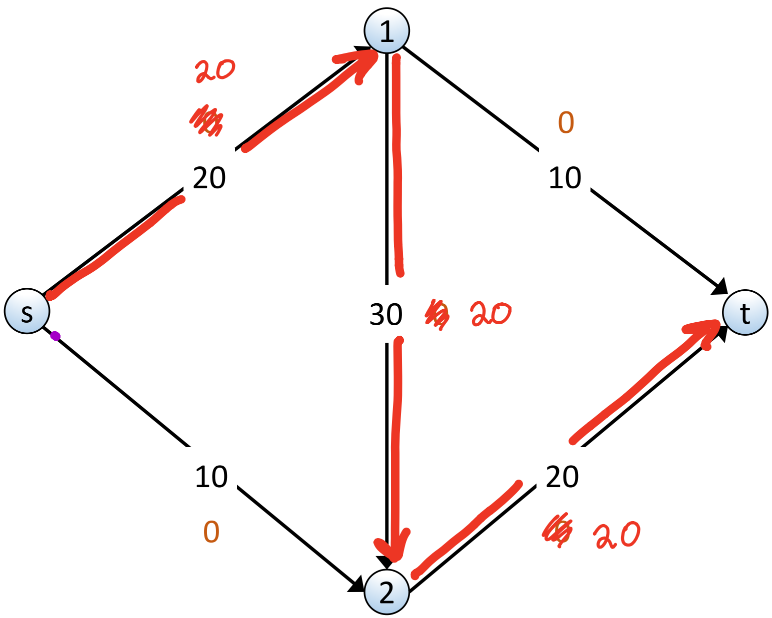
augmenting path

augmenting path

If we add flow along an augmenting path P, all constraints remain satisfied.

Greedy Max Flow

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

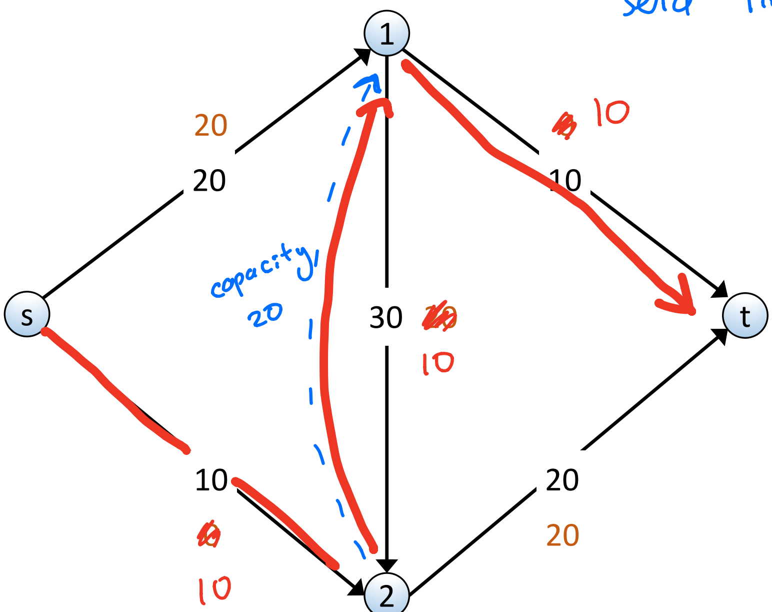


If we choose this augmenting path, we get stuck.

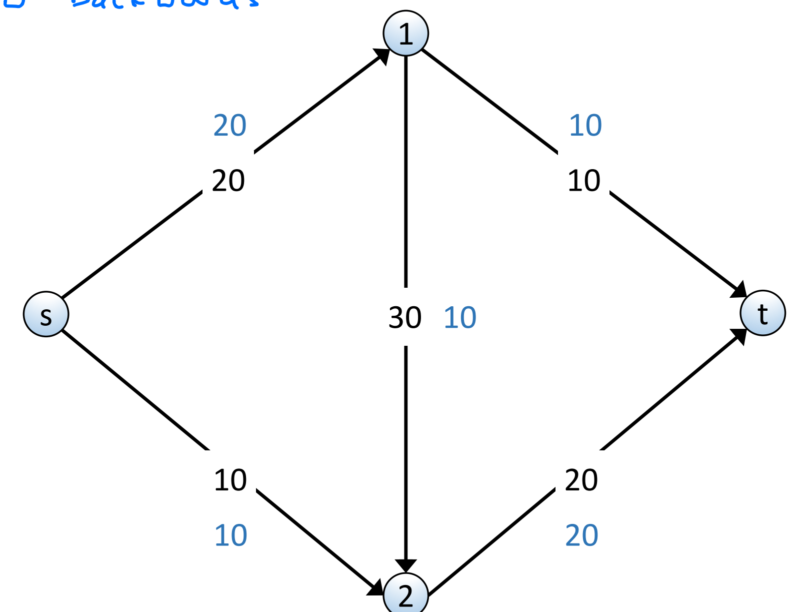
Does Greedy Work?

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

need to cancel flow /
send flow backwards



greedy

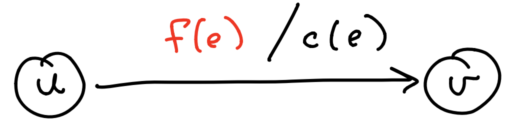


optimal

Residual Graphs

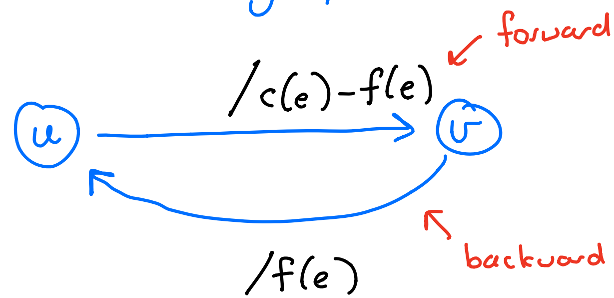
- Original edge: $e = (u, v) \in E$.
  - Flow $f(e)$, capacity $c(e)$

original graph G



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

residual graph G_f



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

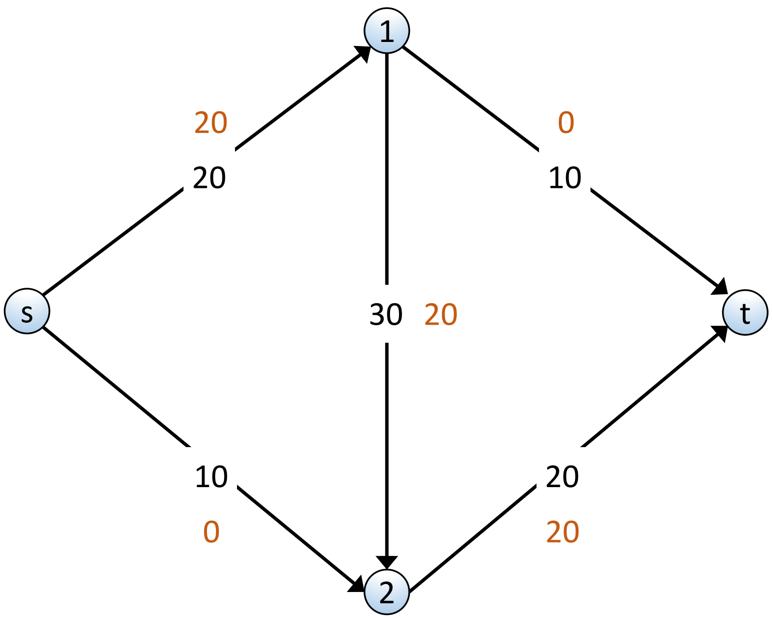
Augmenting Paths in Residual Graphs

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

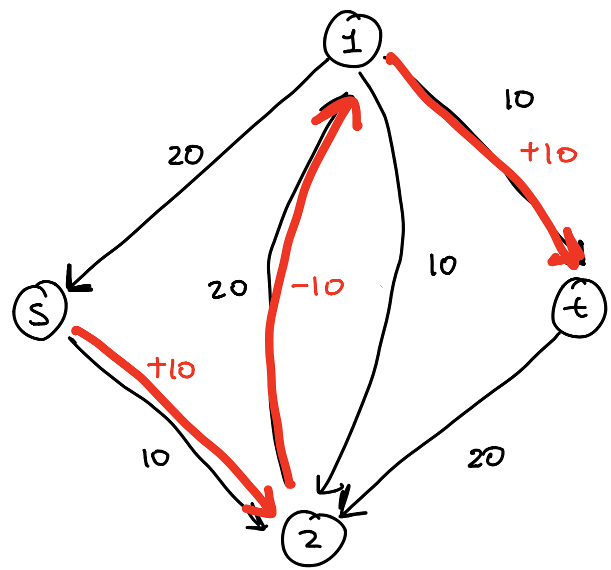
```
Augment(G_f, P)              residual
    b ← the minimum capacity of an edge in P
    for e ∈ P
        if e ∈ E:    f(e) ← f(e) + b
        else:        f(e) ← f(e) - b
    return f
```

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



G_f



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
```

To find one path:

1 Find a path $O(m)$ using BFS

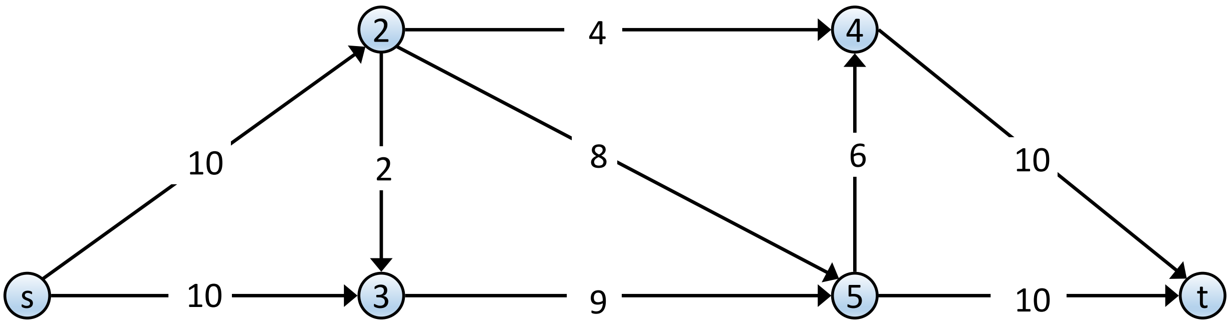
2 Augment $O(n)$ time

3 Update G_f $O(n)$ time

$O(m)$ per augmentation

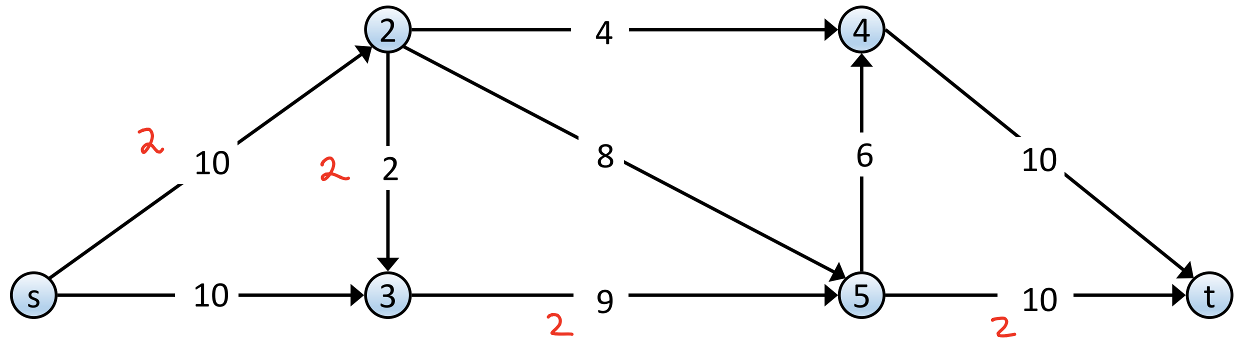
Ford-Fulkerson Demo

G:

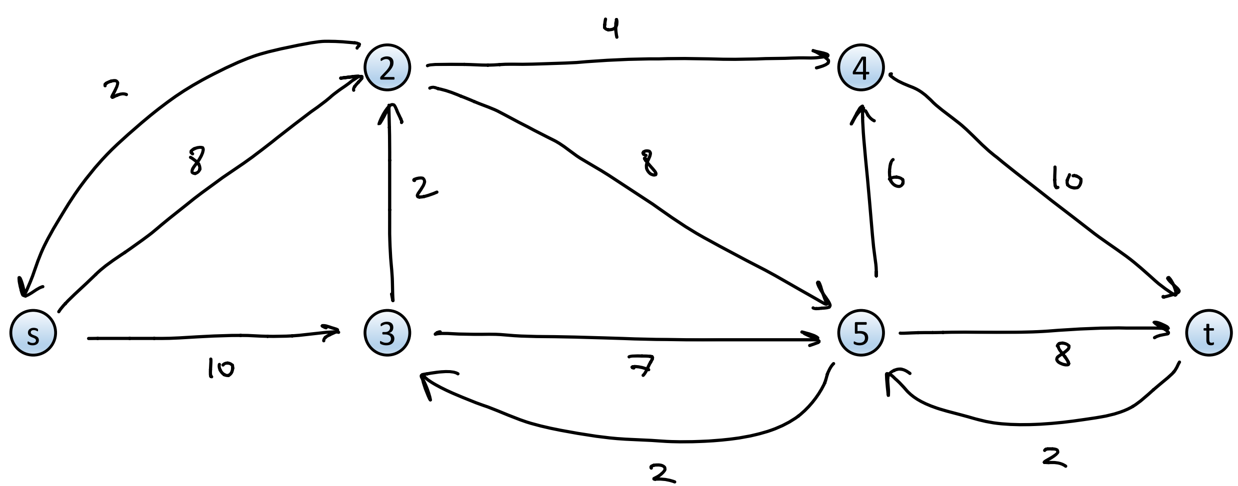


Ford-Fulkerson Demo

G:

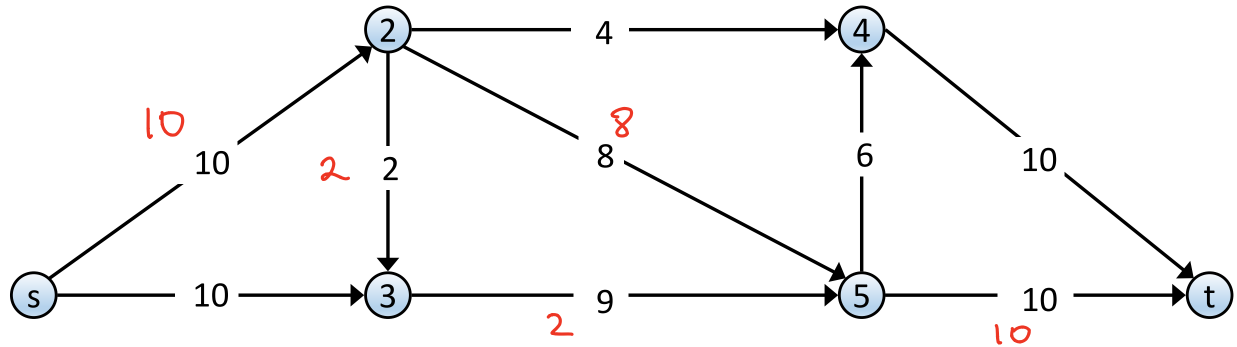


G_f:

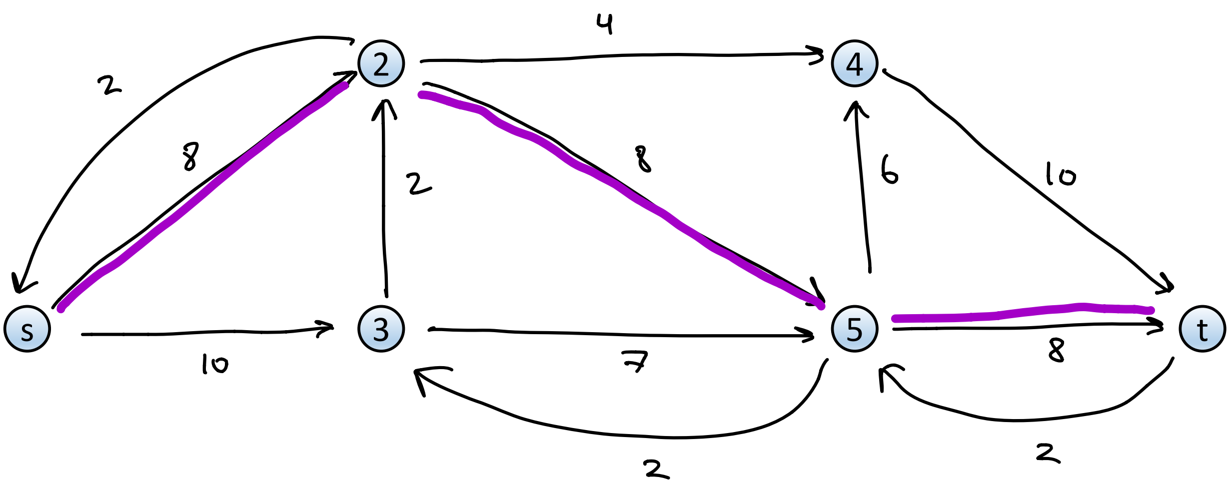


Ford-Fulkerson Demo

G:

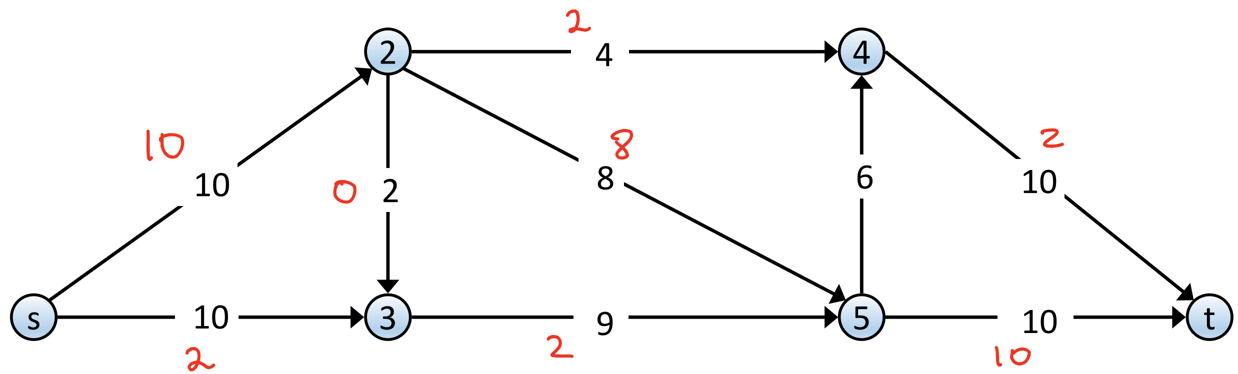


G_f:

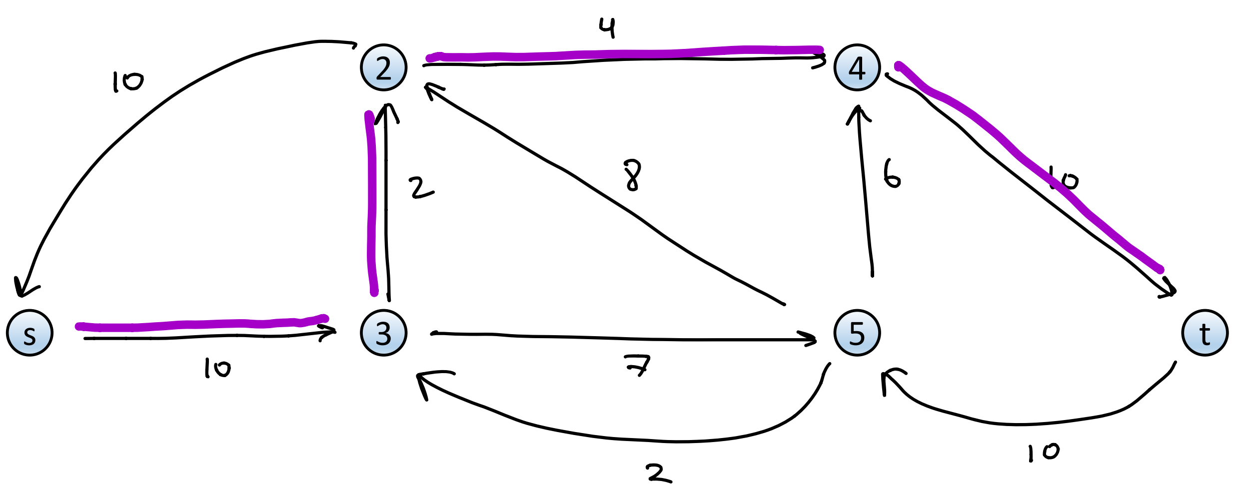


Ford-Fulkerson Demo

G:

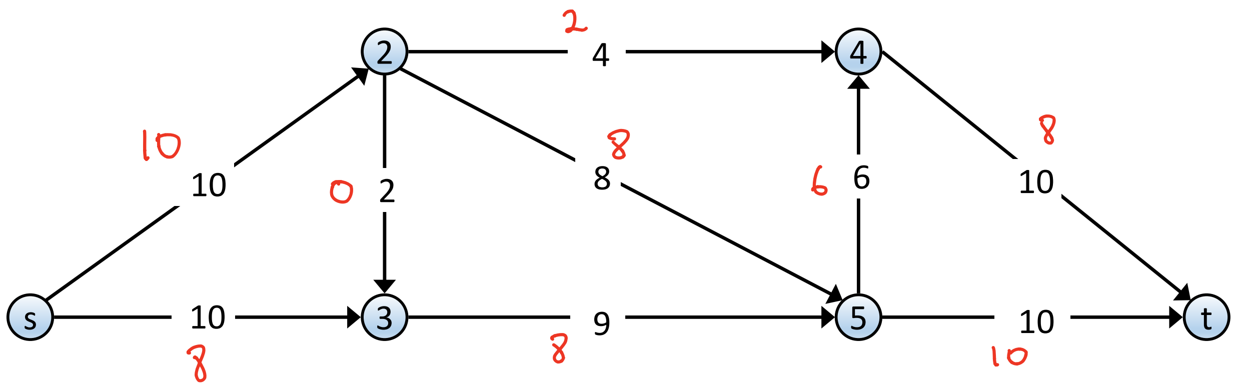


G_f:

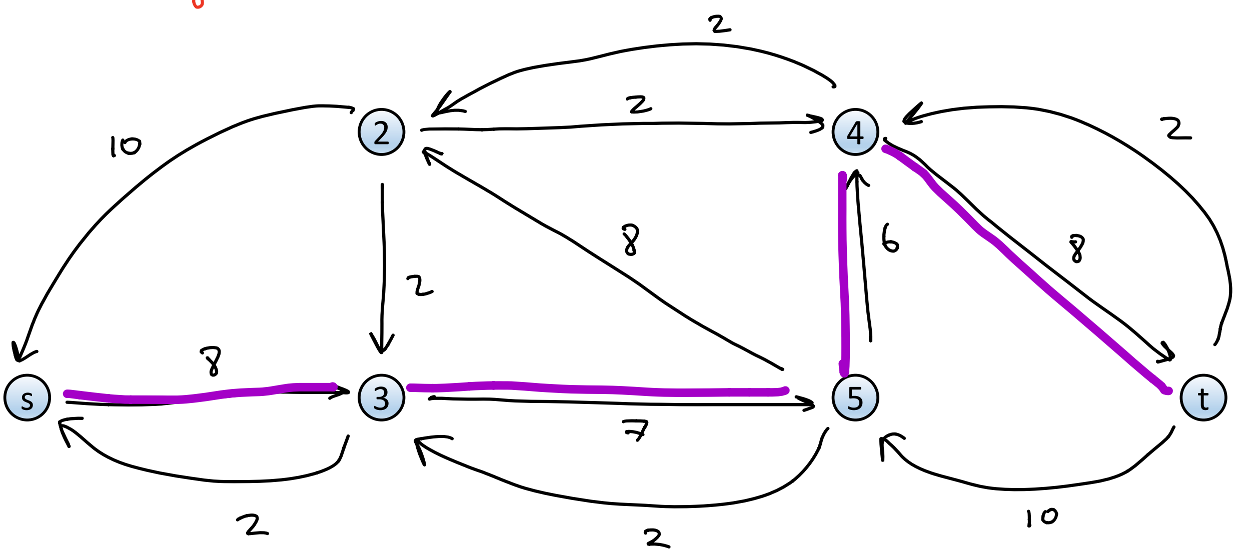


Ford-Fulkerson Demo

G:

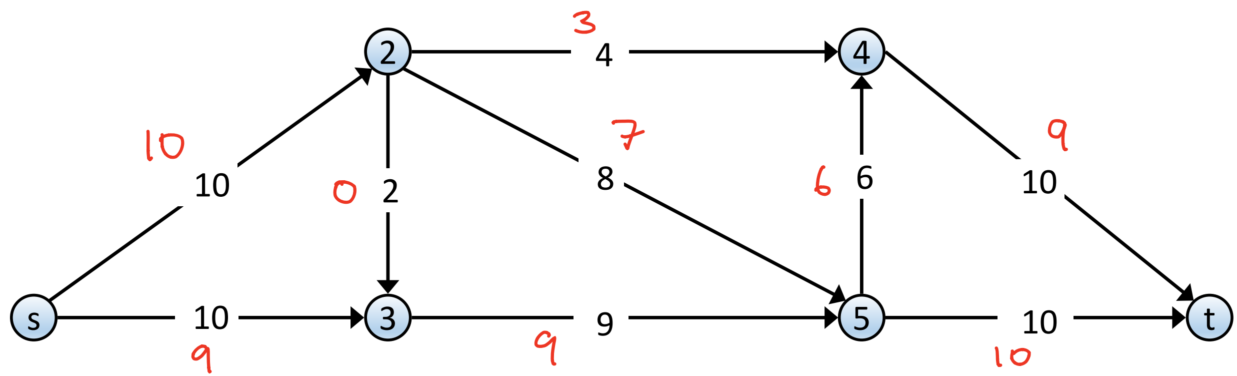


G_f:

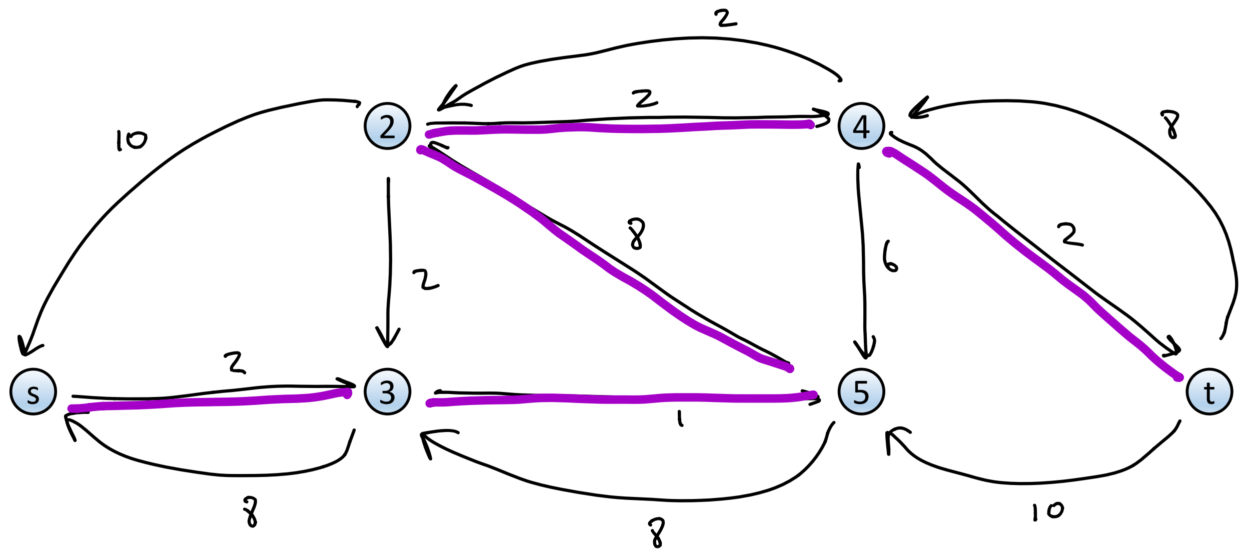


Ford-Fulkerson Demo

G:



G_f:



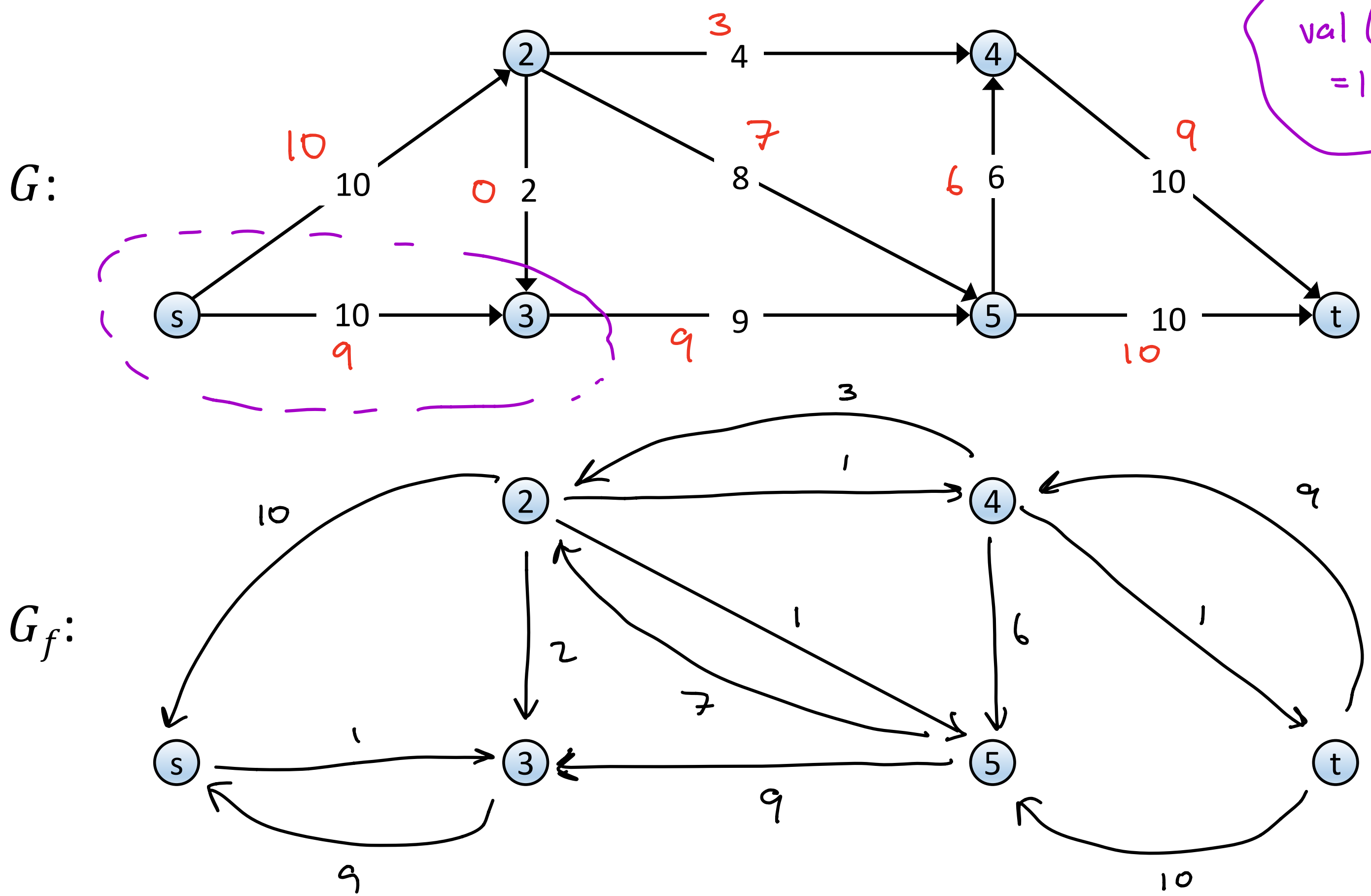
Ford-Fulkerson Demo

A = { s, 3 }

B = { 2, 4, 5, t }

cap(A,B) = 19

val(f) = 19



What do we want to prove?

① FF terminates*

② FF finds a maximum s-t flow

③ There is always a cut (A, B) s.t. $val(f) = cap(A, B)$

Running Time of Ford-Fulkerson

- For **integer capacities**, $\leq val(f^*)$ augmentation steps
  - Every augmentation adds ≥ 1 unit of flow

- Can perform each augmentation step in $O(m)$ time
  - find augmenting path in $O(m)$
  - 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!
  - Can solve max flow in $O(mn)$ time for any capacities

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

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

$(1 \Rightarrow 2)$ By weak duality

$(2 \Rightarrow 3)$ If there is an augmenting path the f can't be maximum

Hard Part: $(3 \Rightarrow 1)$

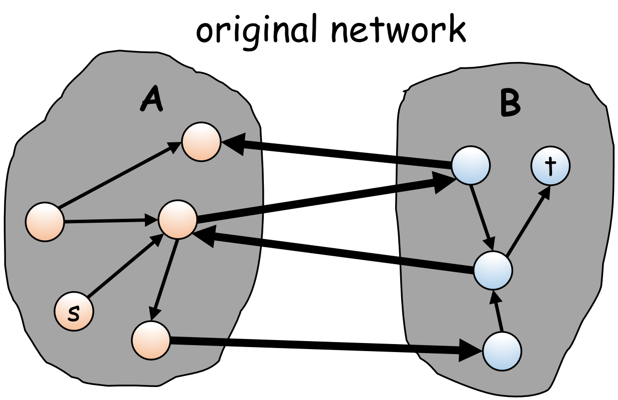
Optimality of Ford-Fulkerson

- **(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

(A, B) is an s-t cut because $t \notin A$ (otherwise t is reachable from A)

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



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

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

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

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

Summary

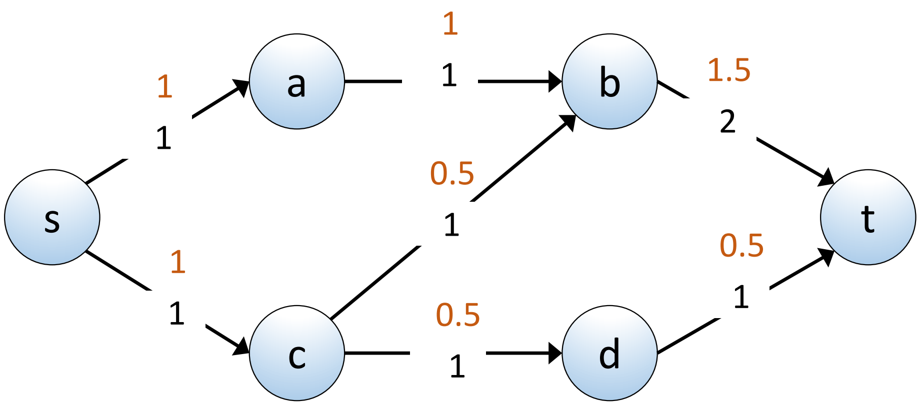
- **The Ford-Fulkerson Algorithm solves maximum s-t flow**
  - Running time $O\big(m \cdot val(f^*)\big)$ in networks with integer capacities
  - Space $O(n + m)$

- **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 integer capacities has an integer maximum flow**
  - Ford-Fulkerson will return an integral maximum flow

Ask the Audience

- Is this a maximum flow?



- Is there an **integer maximum flow**?
- Does every graph with integer capacities have an integer 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 integer capacities has an integer maximum flow**
  - Ford-Fulkerson will return an integer maximum flow