Algorithm Design and Analysis

CSE 565

LECTURES 17

Network Flow

- Duality of Max Flow **and** Min Cut
- Algorithms:
  - Ford-Fulkerson
  - Capacity Scaling

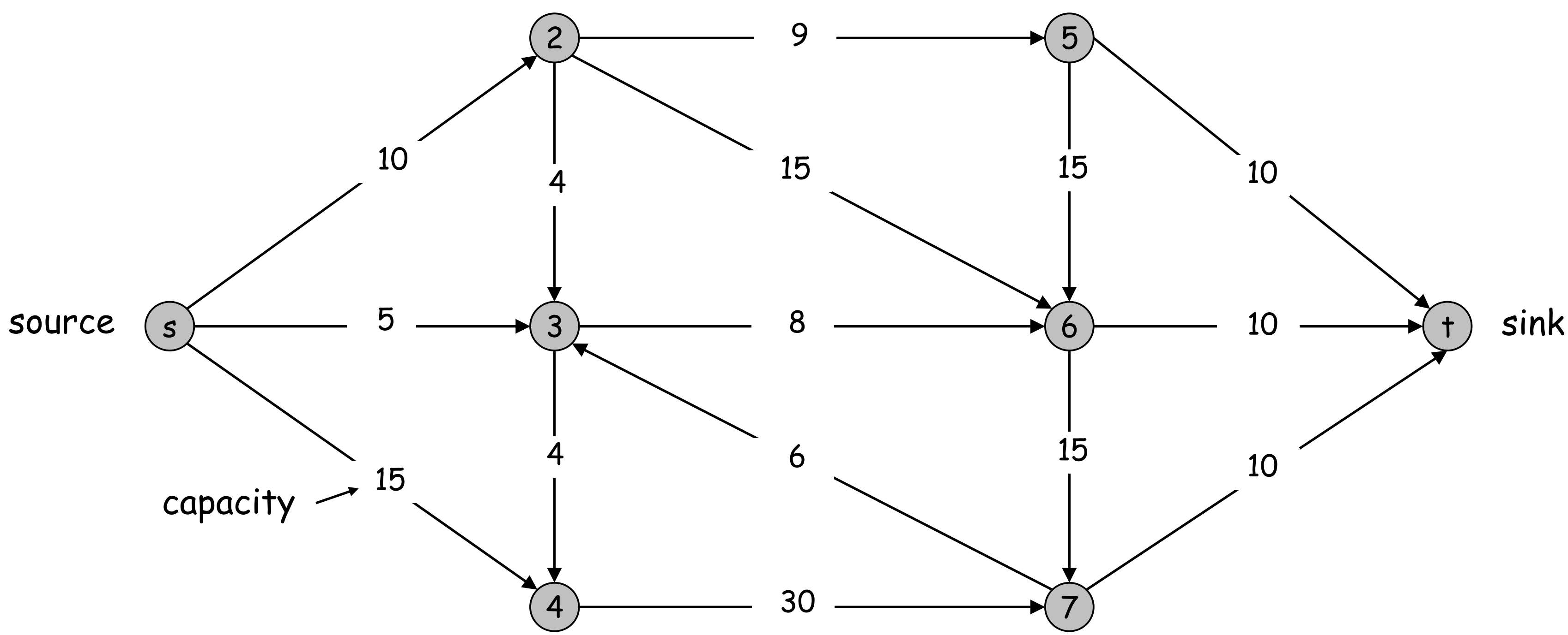
Sofya Raskhodnikova

Network Flow

Minimum Cut Problem

Flow network.

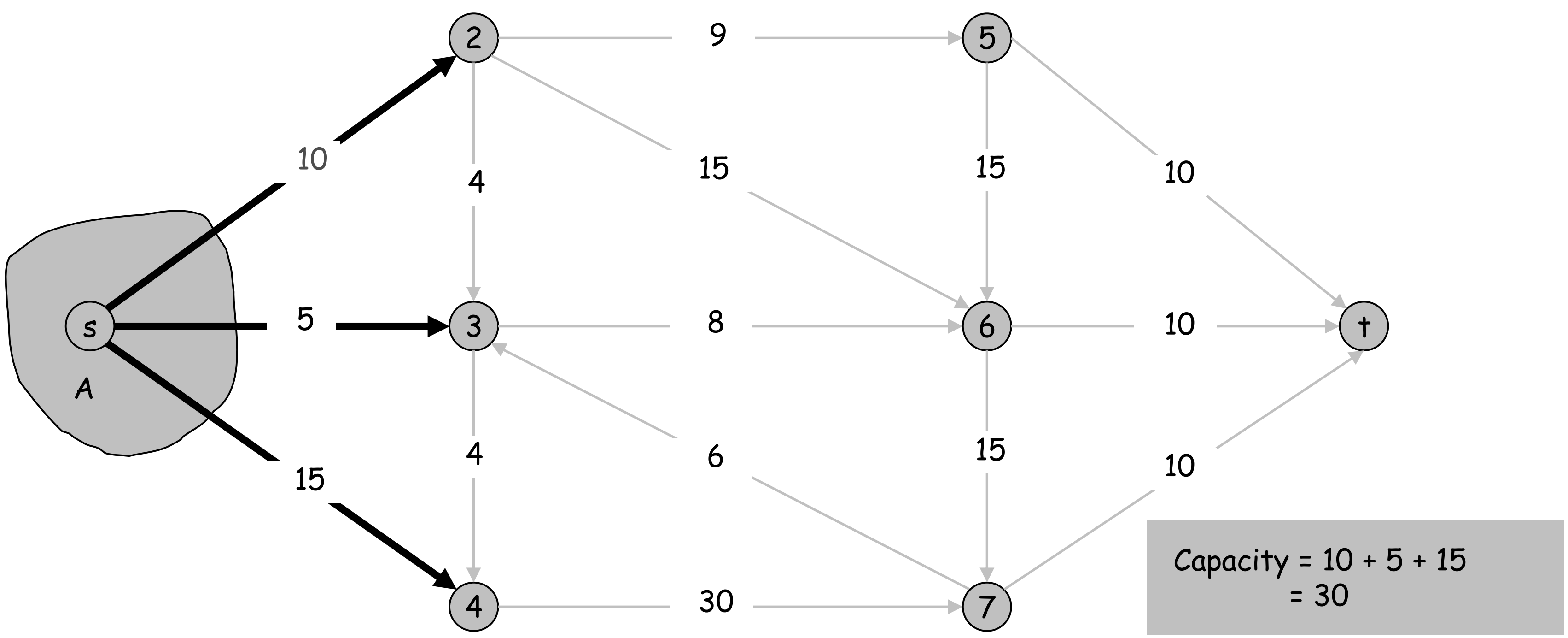
- Abstraction for material flowing through the edges.
- G = (V, E) = directed graph, no parallel edges.
- Two distinguished nodes: s = source, t = sink.
- c(e) = capacity of edge e.



Cuts

Def. An **s-t cut** is a partition (A, B) of V with s ∈ A and t ∈ B.

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

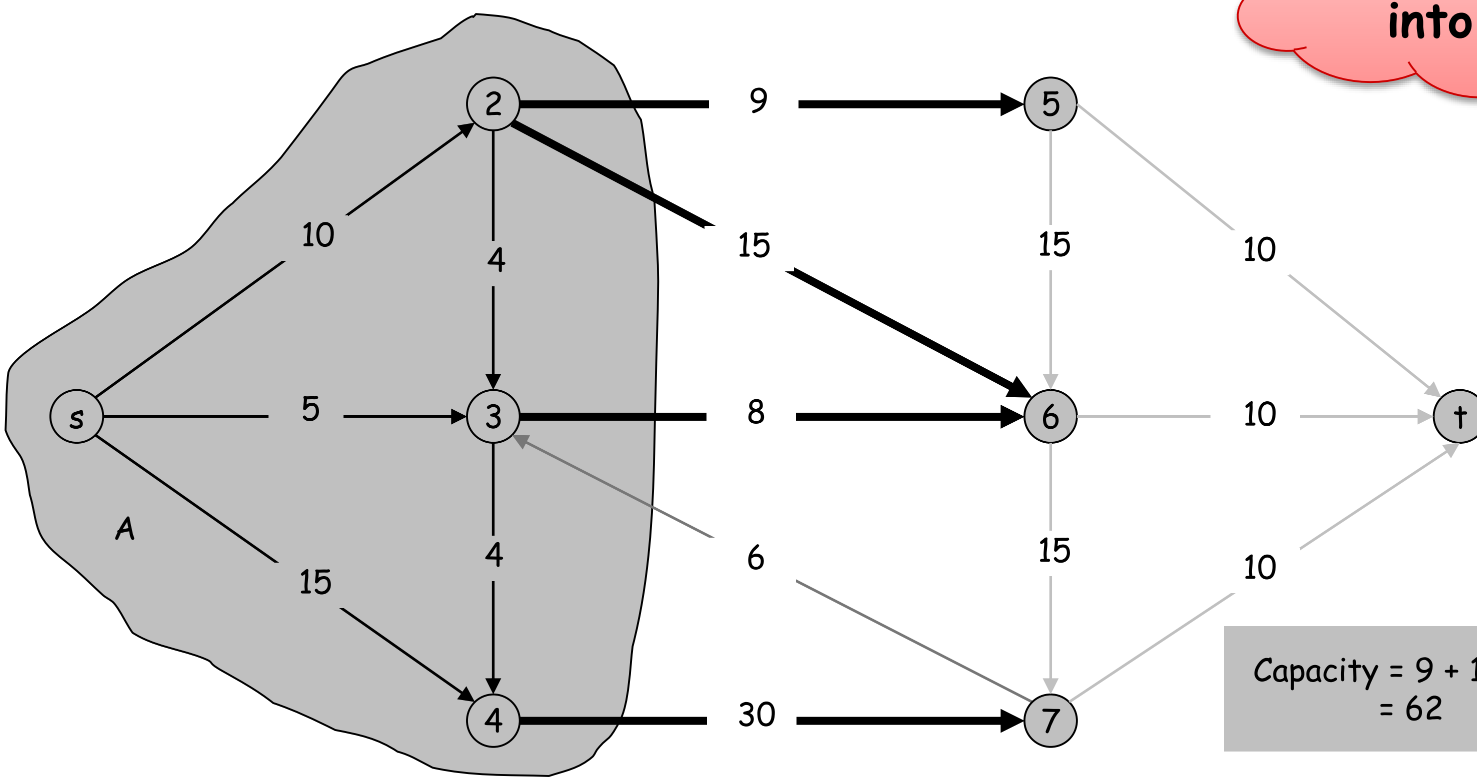


Cuts

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

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

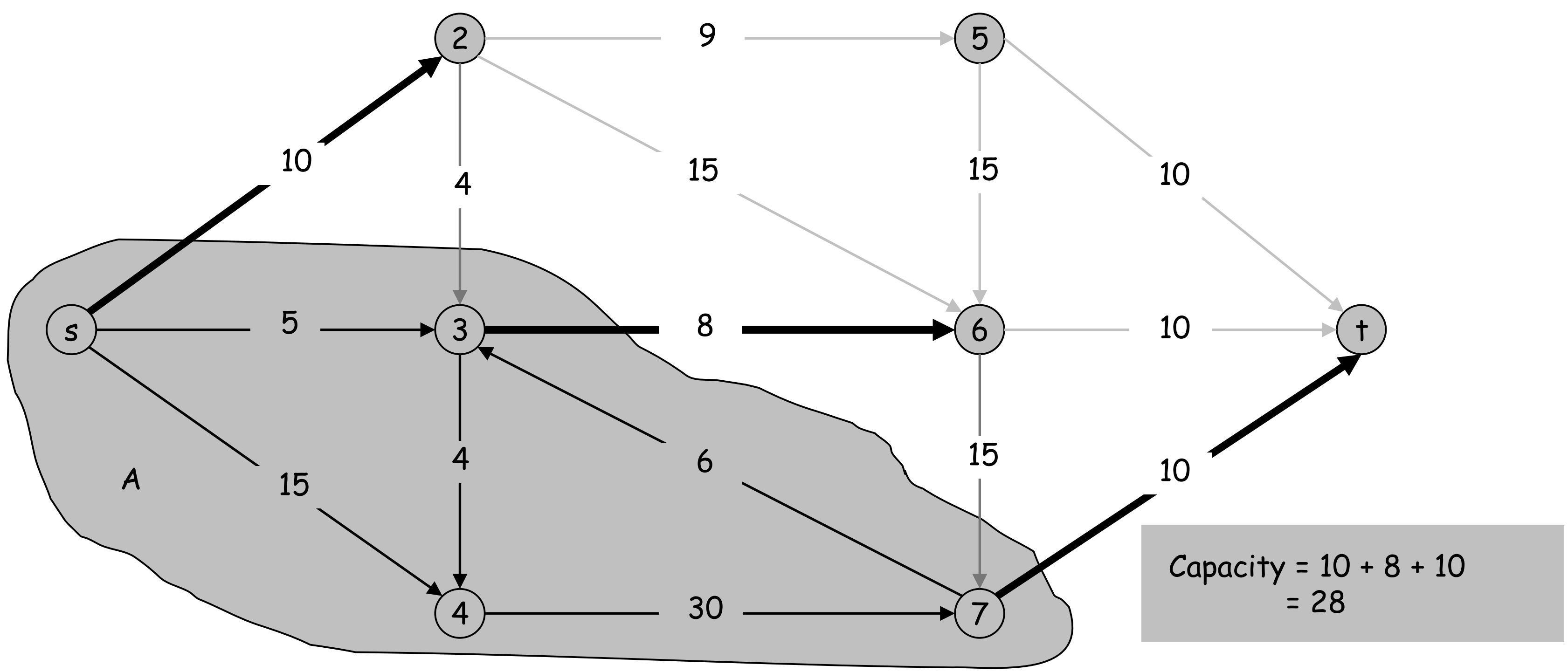
we don't count edges **into** A



Capacity = 9 + 15 + 8 + 30
= 62

Minimum Cut Problem

Min s-t cut problem. Find an s-t cut of minimum capacity.



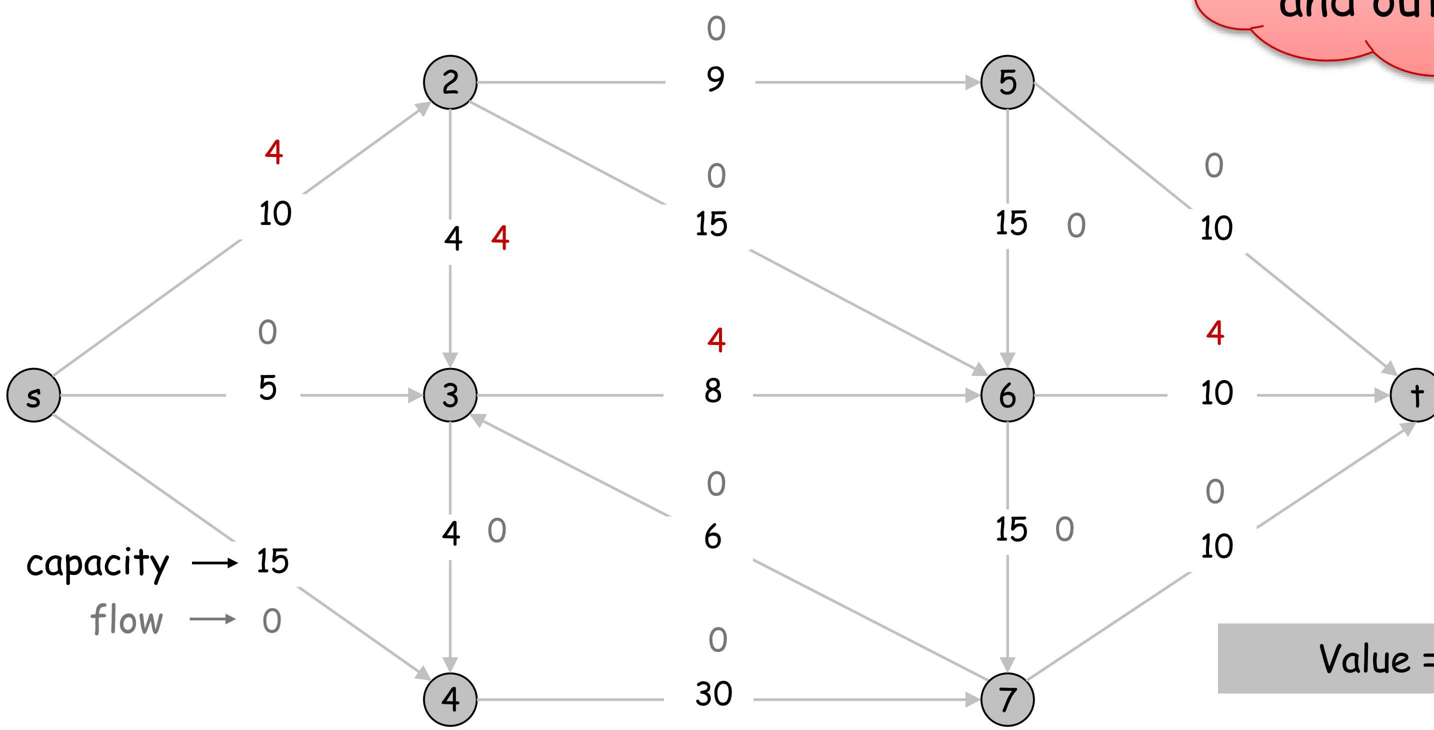
Flows

Def. An s-t flow is a function that satisfies:

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

Water flowing in and out of v

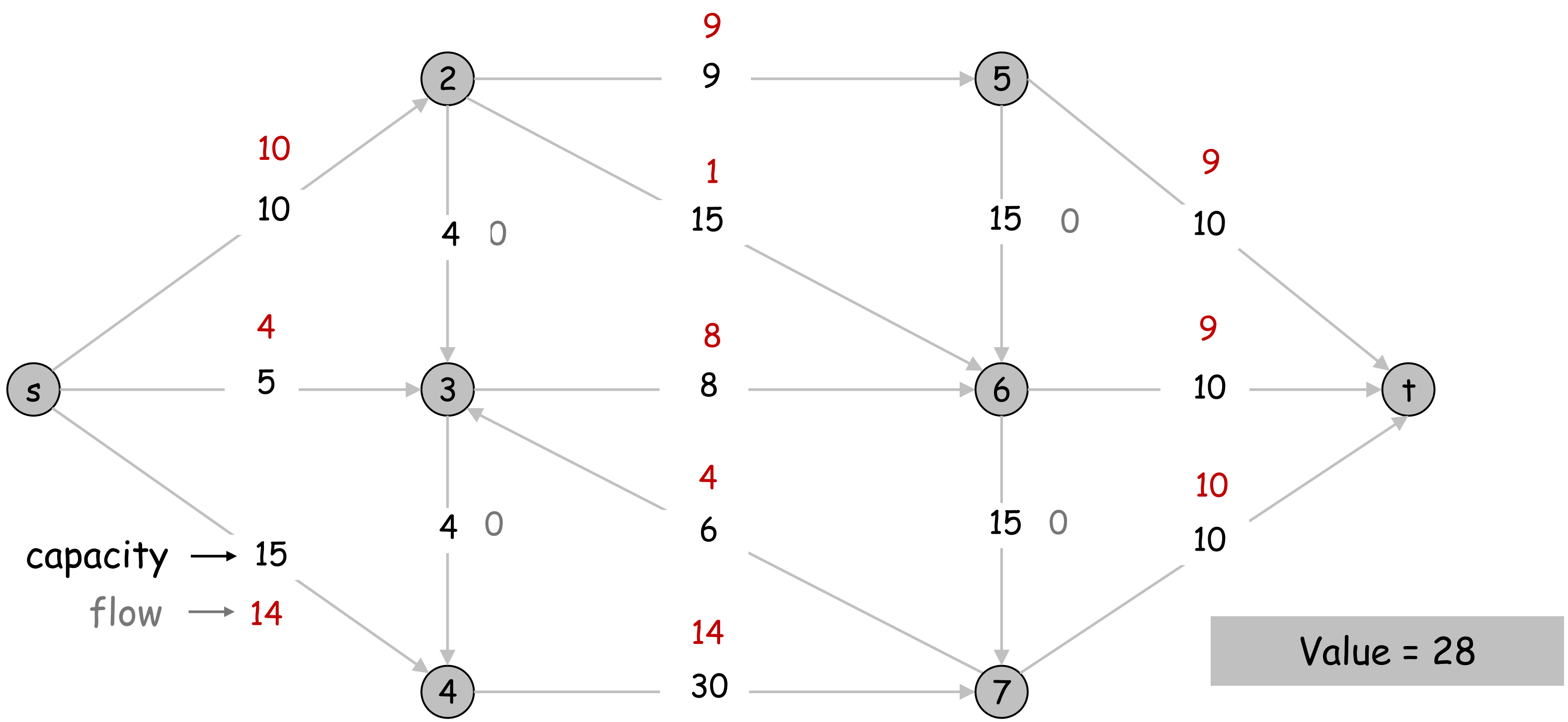
Def. The value of a flow f is: $\quad v(f) = \sum_{e \text{ out of } s} f(e)$.



Value = 4

Maximum Flow Problem

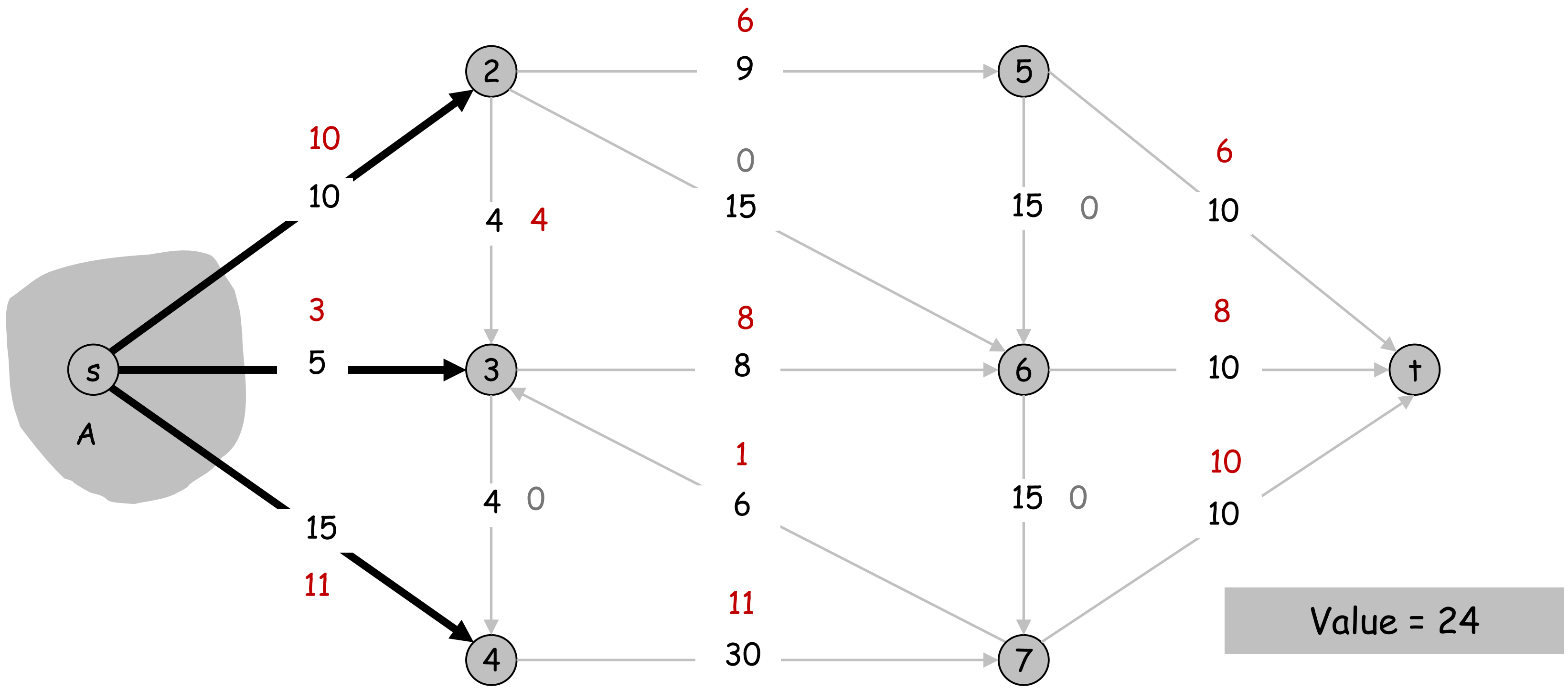
Max flow problem. Find s-t flow of maximum value.



Flows and Cuts

Flow value lemma. Let f be any flow, and let (A, B) be any s-t cut. Then the net flow sent across the cut is equal to the amount leaving s.

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



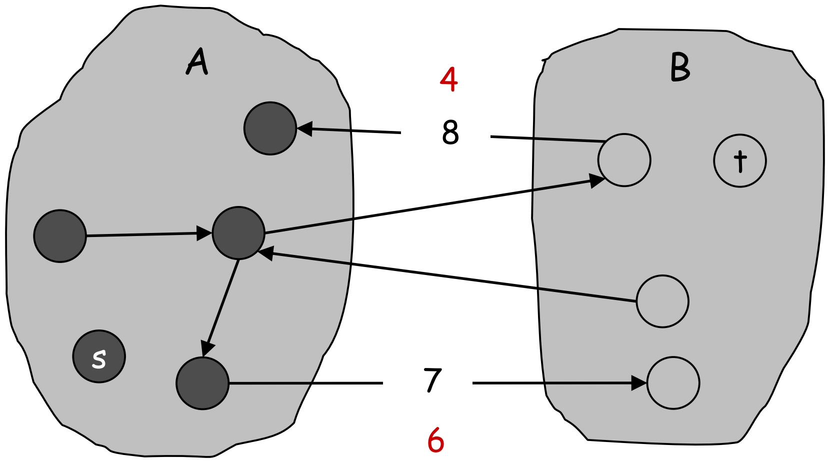
Flows and Cuts

Weak duality. Let f be any flow. Then, for any s-t cut (A, B),

$$v(f) \le cap(A, B).$$

Proof.

$$\begin{aligned} v(f) &= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) \\ &\le \sum_{e \text{ out of } A} f(e) \\ &\le \sum_{e \text{ out of } A} c(e) \\ &= \text{cap}(A, B) \quad \blacksquare \end{aligned}$$



Review Questions

True/False

Let G be an arbitrary flow network, with a source s, and sink t, and a positive integer capacity c_e on every edge e.

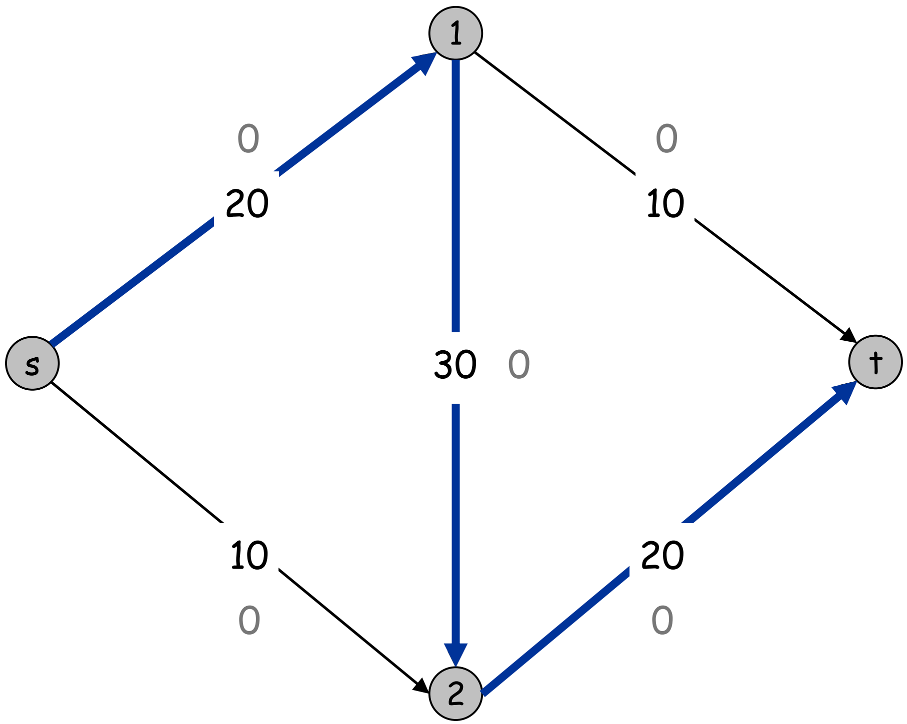
1) If f is a maximum s-t flow in G, then f saturates every edge out of s with flow (i.e., for all edges e out of s, we have $f(e) = c_e$).

2) Let (A,B) be a minimum s-t cut with respect to these capacities . If we add 1 to every capacity, then (A,B) is still a minimum s-t cut with respect to the new capacities.

Towards a Max Flow Algorithm

Greedy algorithm.

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

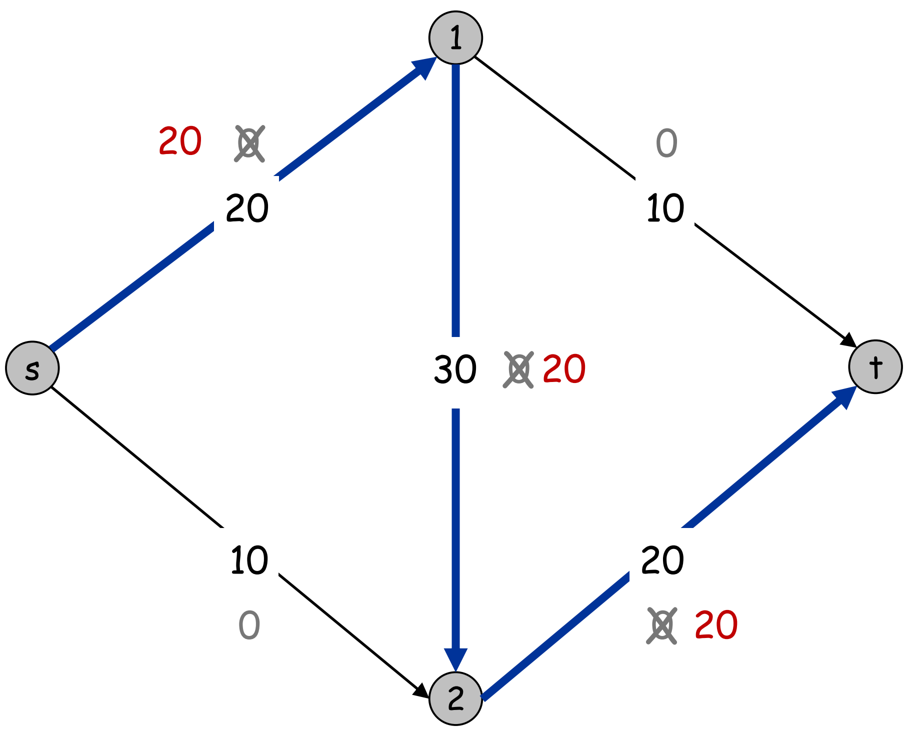


Flow value = 0

Towards a Max Flow Algorithm

Greedy algorithm.

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



Flow value = 20

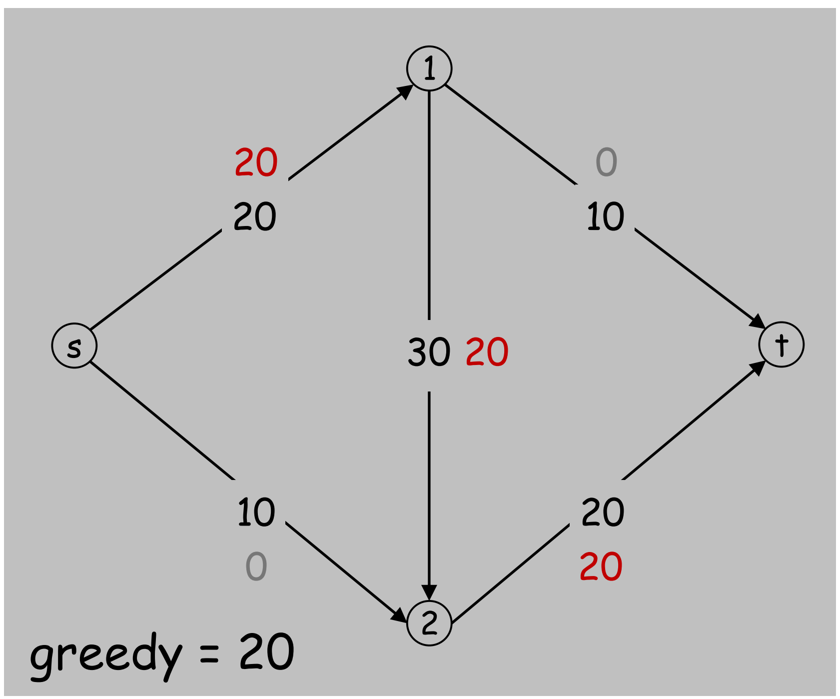
Towards a Max Flow Algorithm

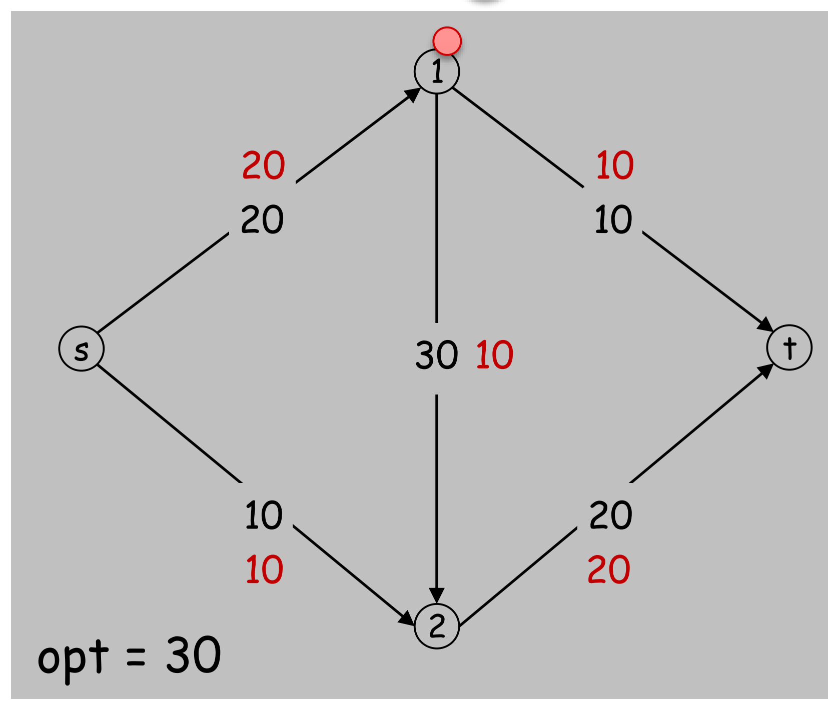
How can we get from greedy to opt here? What if we **push water back** across middle edge?

Greedy algorithm.

- Start with f(e) = 0 for all edge e ∈ E.
- Find an s-t path P where each edge has f(e) < c(e).
- Augment flow along path P.
- Repeat until you get stuck.

↖ locally optimality ⇏ global optimality

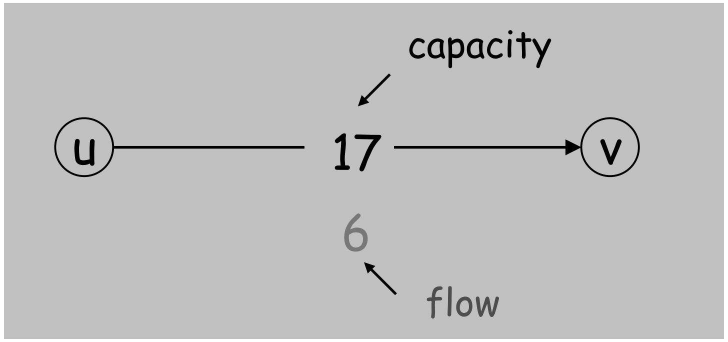




Residual Graph

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

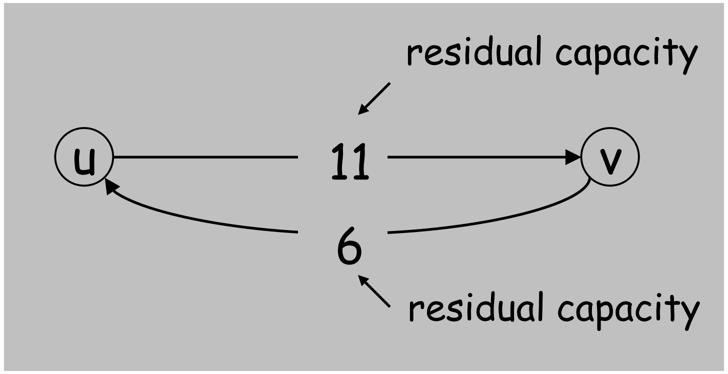
- Flow $f(e)$, capacity $c(e)$.



Residual edge.

- "Undo" flow sent.
- $e = (u, v)$ and $e^R = (v, u)$.
- Residual capacity:

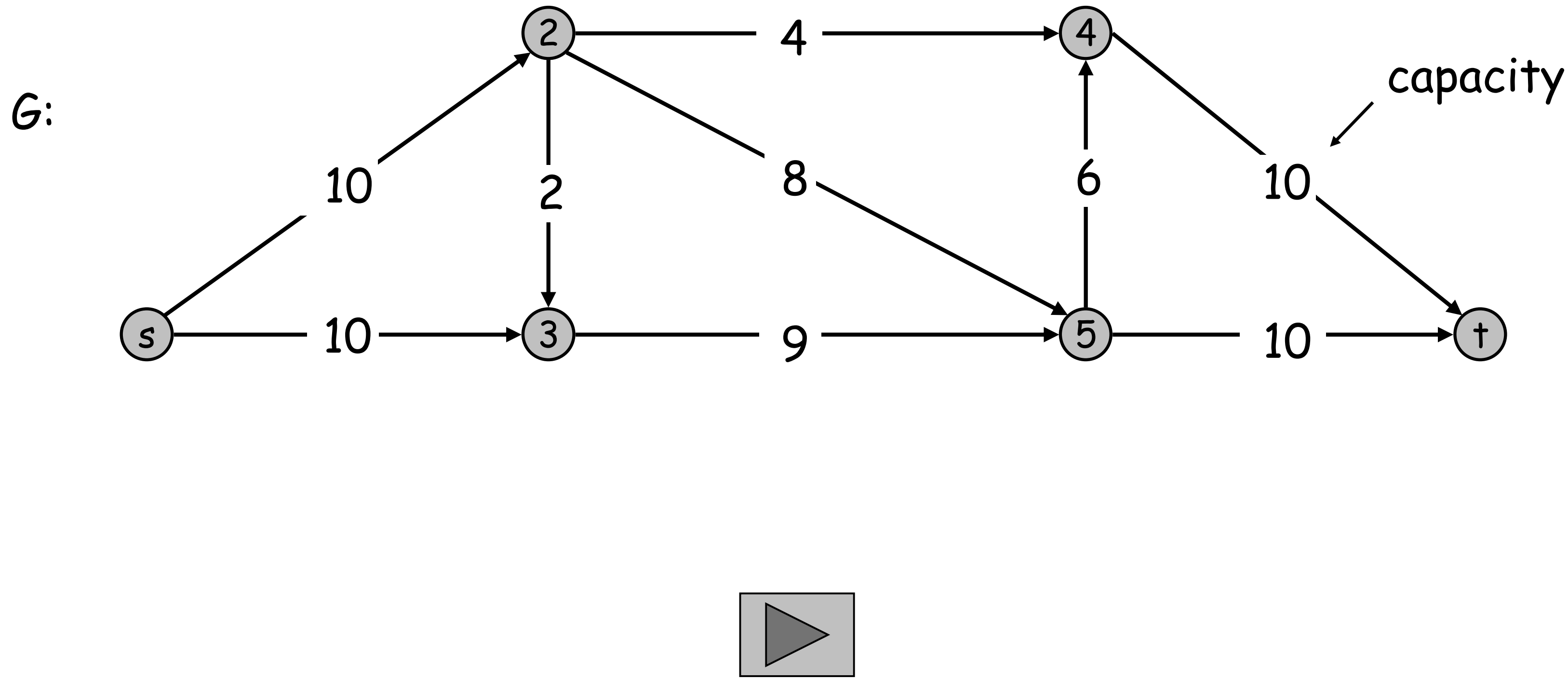
$$c_f(e) = \begin{cases} c(e) - f(e) & \text{if } e \in E \\ f(e) & \text{if } e^R \in E \end{cases}$$



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

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

Ford-Fulkerson Algorithm



Augmenting Path Algorithm

```
Ford-Fulkerson(G, s, t, c) {
   foreach e ∈ E,  f(e) ← 0
   Gf ← residual graph

   while (there exists augmenting path P) {
      f ← Augment(f, c, P)
      update Gf
   }
   return f
}
```

```
Augment(f, c, P) {
   b ← bottleneck-capacity(P)
   foreach e ∈ P {
      if (e ∈ E) f(e) ← f(e) + b
      else       f(e^R) ← f(e^R) - b
   }
   return f
}
```

Min residual capacity of an edge in P

forward edge

reverse edge

Ford-Fulkerson: Analysis

Ford-Fulkerson summary:

- **While** you can,
  - Greedily push flow
  - Update residual graph

Lemma 1 (Feasibility): Ford-Fulkerson outputs a valid flow.
Proof: Check capacity and conservation conditions. (Details in KT)

Still to do:

- Optimality: Does it output a **maximum** flow?
- Running time (in particular, termination!)

Max-Flow Min-Cut Theorem

Augmenting path theorem. Flow f is a max flow iff there are no augmenting paths.

Max-flow min-cut theorem. [Ford-Fulkerson 1956] The value of the max flow is equal to the value of the min cut.

Proof strategy. We prove both simultaneously by showing that TFAE:

(i) There exists a cut (A, B) such that v(f) = cap(A, B).

(ii) Flow f is a max flow.

(iii) There is no augmenting path relative to f.

(i) $\Rightarrow$ (ii) This was the corollary to weak duality lemma.

(ii) $\Rightarrow$ (iii) We show contrapositive.

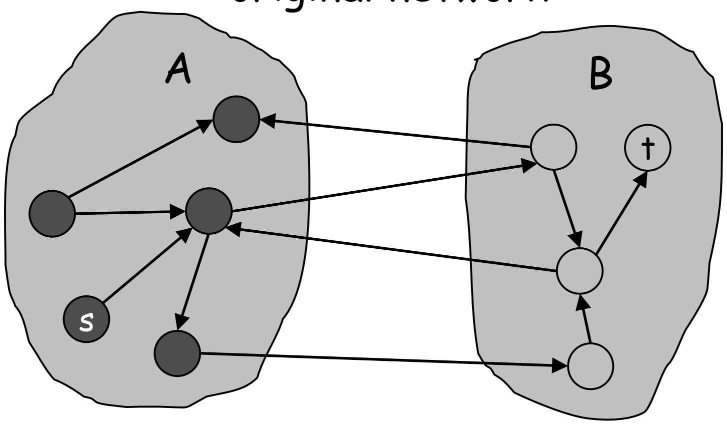
- Let f be a flow. If there exists an augmenting path, then we can improve f by sending flow along path.

Proof of Max-Flow Min-Cut Theorem

(iii) $\Rightarrow$ (i)

- Let f be a flow with no augmenting paths.
- Let A be set of vertices reachable from s in residual graph.
- By definition of A, source s $\in$ A.
- By definition of f, sink t $\notin$ A.
- Observation: No edges of the residual graph go from A to B.

original network



- **Claim 1**: If e goes from A to B, then f(e) =c(e).
  Proof: Otherwise there would be residual capacity, and the residual graph would have an edge A to B.

- **Claim 2**: If e goes from B to A, then f(e)=0.
  Proof: Otherwise residual edge would go from A to B.

$$\begin{aligned} v(f) &= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to A}} f(e) \\ &= \sum_{e \text{ out of } A} c(e) \\ &= cap(A, B) \quad \blacksquare \end{aligned}$$

Ford-Fulkerson: Analysis

Ford-Fulkerson summary:

- **While** you can,
  - Greedily push flow
  - Update residual graph

Lemma 1 (Feasibility): Ford-Fulkerson outputs a valid flow.

Augmenting path theorem. Flow f is a max flow iff there are no augmenting paths.

Optimality: If Ford-Fulkerson terminates then the output is a max flow.

Still to do:

- Running time (in particular, termination!)