CSE 421

Max Flow Problem

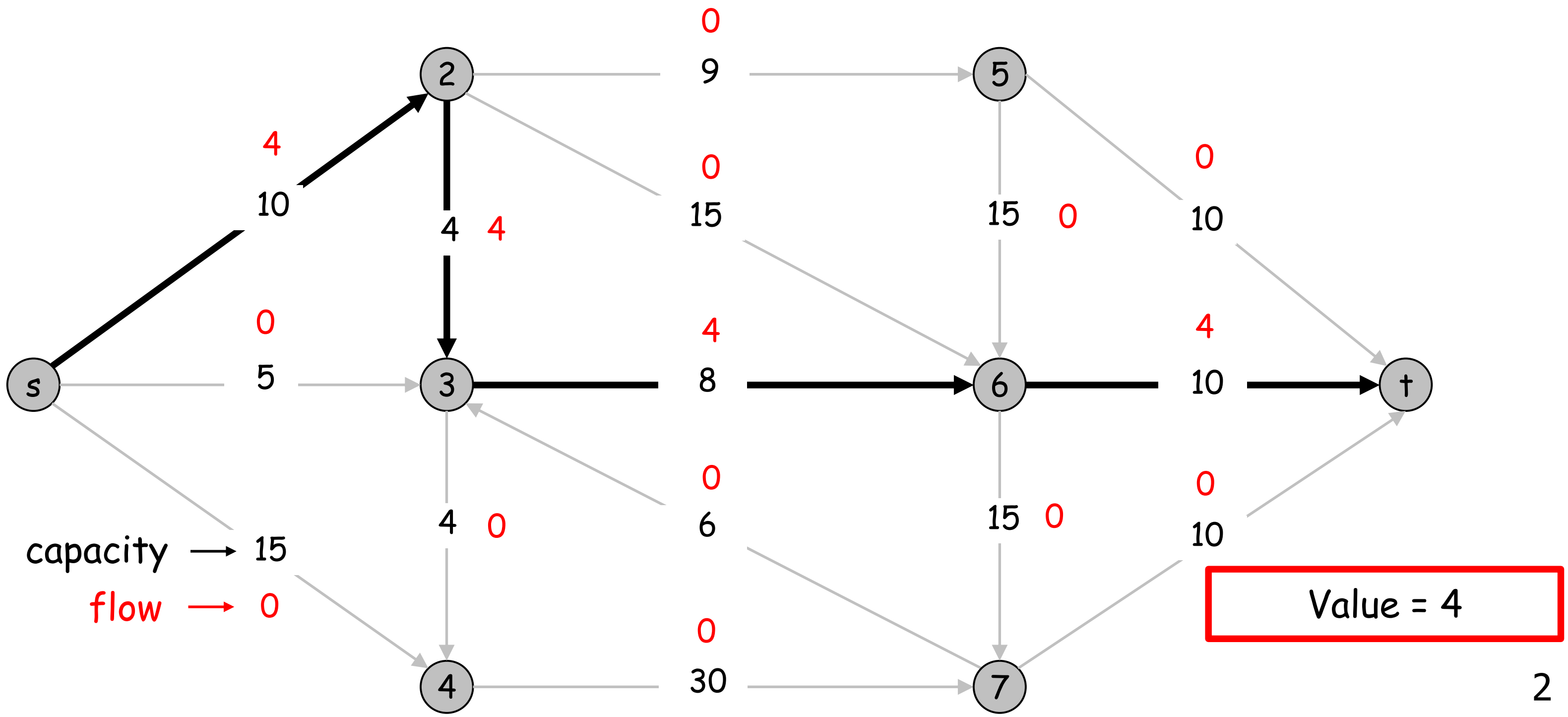
Yin Tat Lee

Last Lecture: s-t Flows

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

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

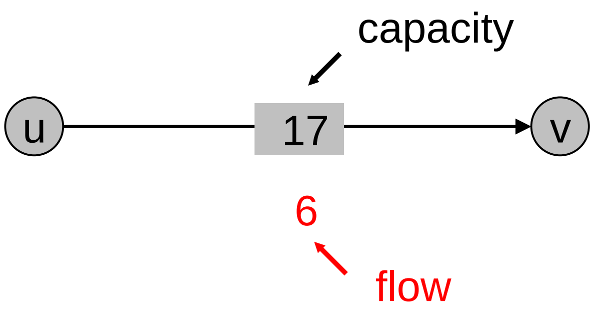
Def. The value of a flow f is: $v(f) = \sum_{e \text{ out of } s} f(e)$



Last Lecture: Residual Graph

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

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

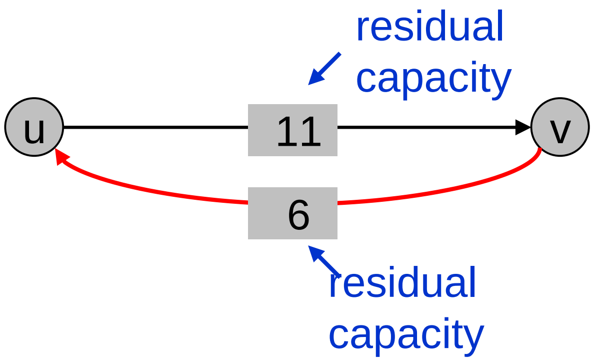


Residual edge.

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

Residual capacity:

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



Residual graph: G_f = (V, E_f).

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

Last Lecture: Augmenting Path Algorithm

```
Augment(f, c, P) {
   b ← bottleneck(P)        ← Smallest capacity edge on P
   foreach e ∈ P {
      if (e ∈ E) f(e) ← f(e) + b     ← Forward edge
      else       f(eR) ← f(e) - b    ← Reverse edge
   }
   return f
}
```

```
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
}
```

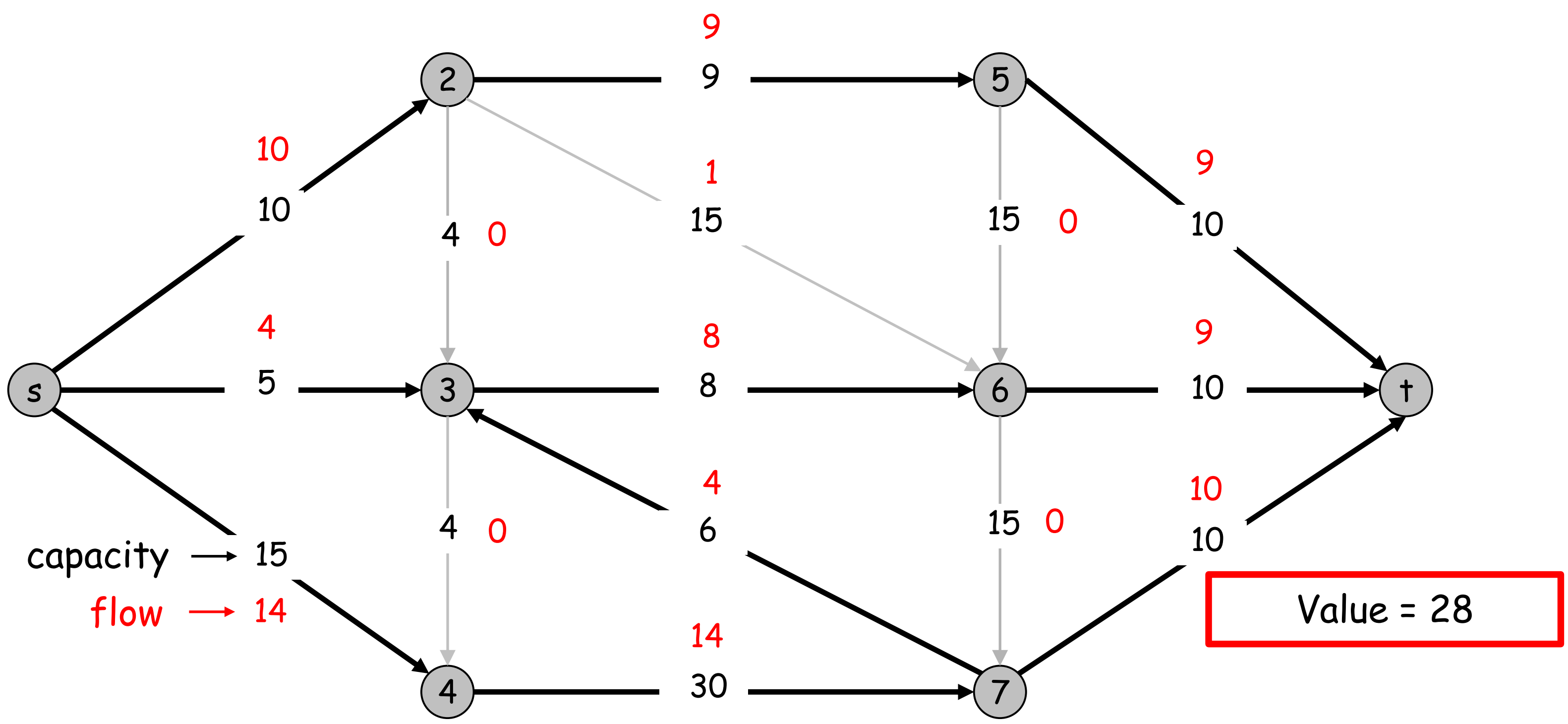
Question:
How to check if a flow is maximum

Outline

- Discuss how to prove a flow is optimal
  - Introduce it as a new problem
- Relate two problems
- Prove correctness of augmenting path
- Some exercise

Maximum s-t Flow Problem

Exercise: Is this a maximum flow?



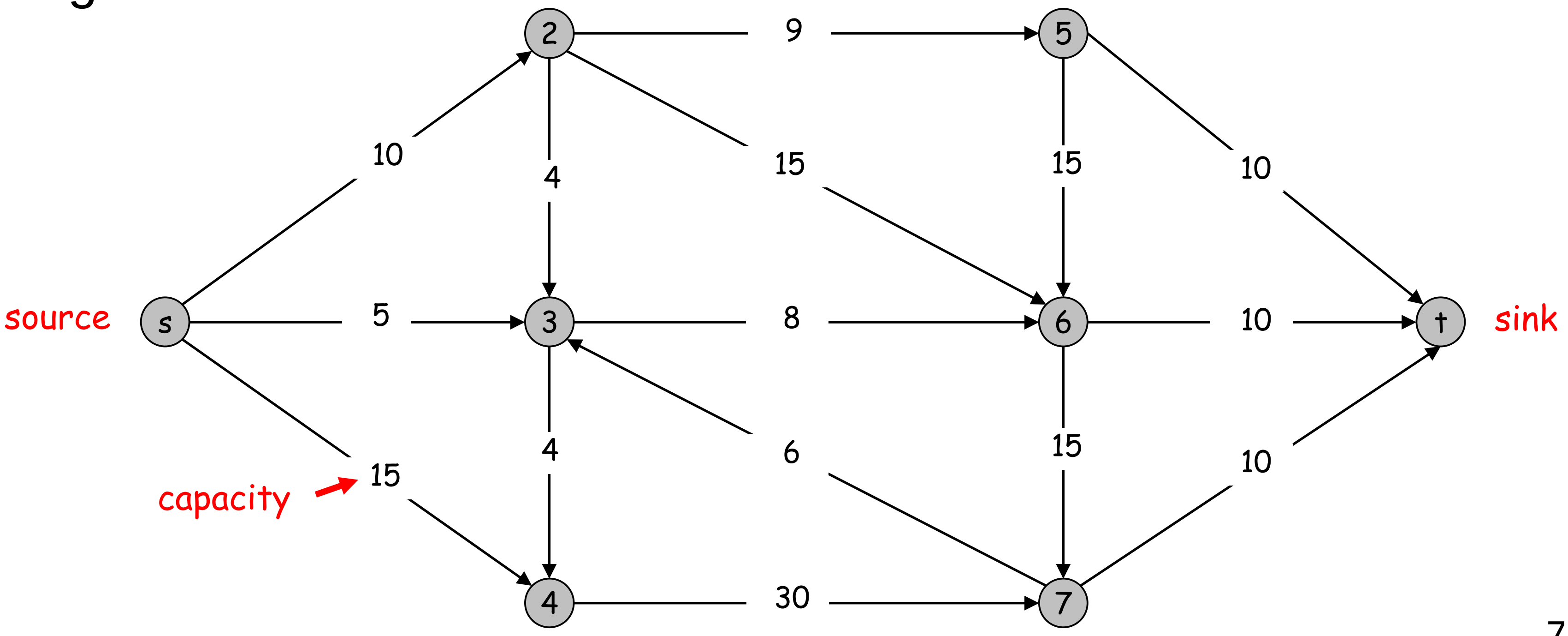
We can verify a maxflow via a cut.

Minimum s-t Cut Problem

Given a directed graph G = (V, E) = directed graph and two distinguished nodes: s = source, t = sink.

Suppose each directed edge e has a nonnegative capacity $c(e)$

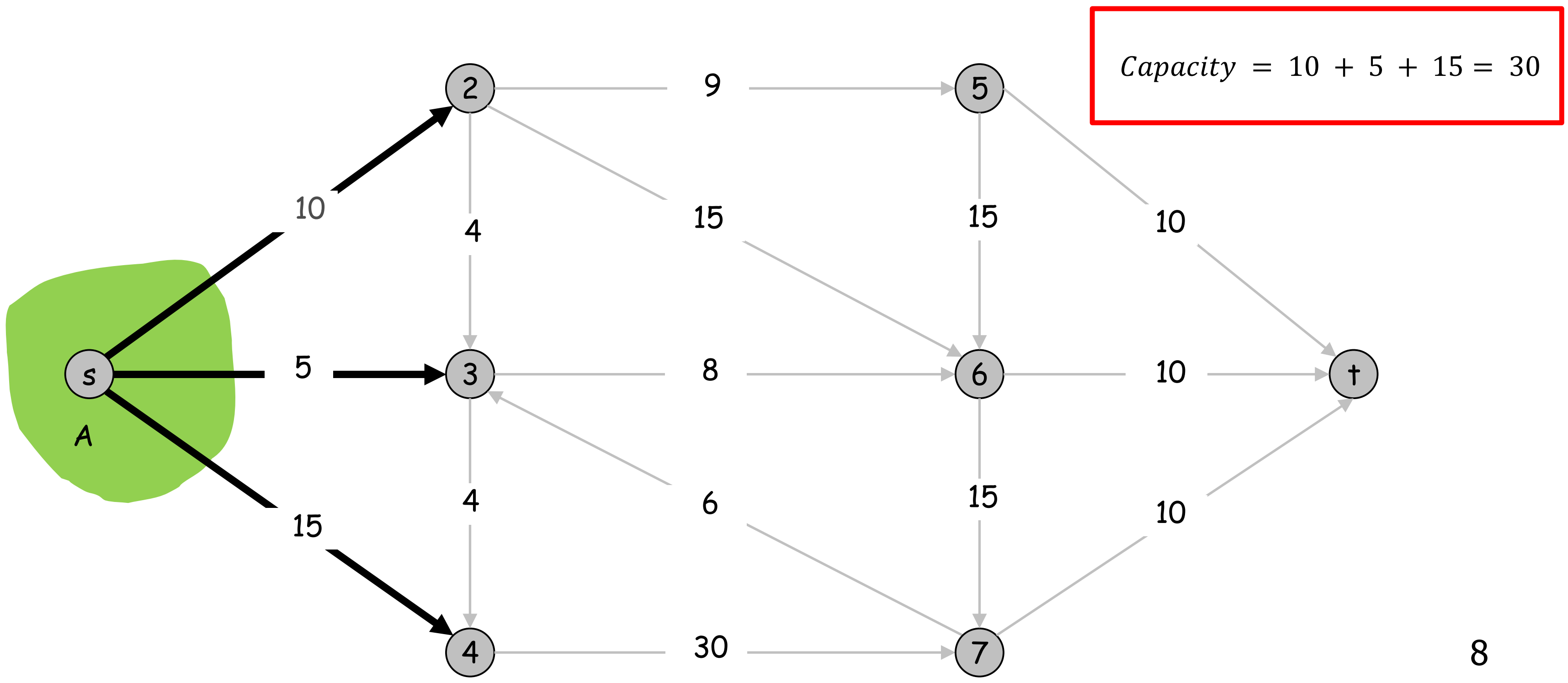
Goal: Find a cut separating s, t that cuts the minimum capacity of edges.



s-t cuts

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

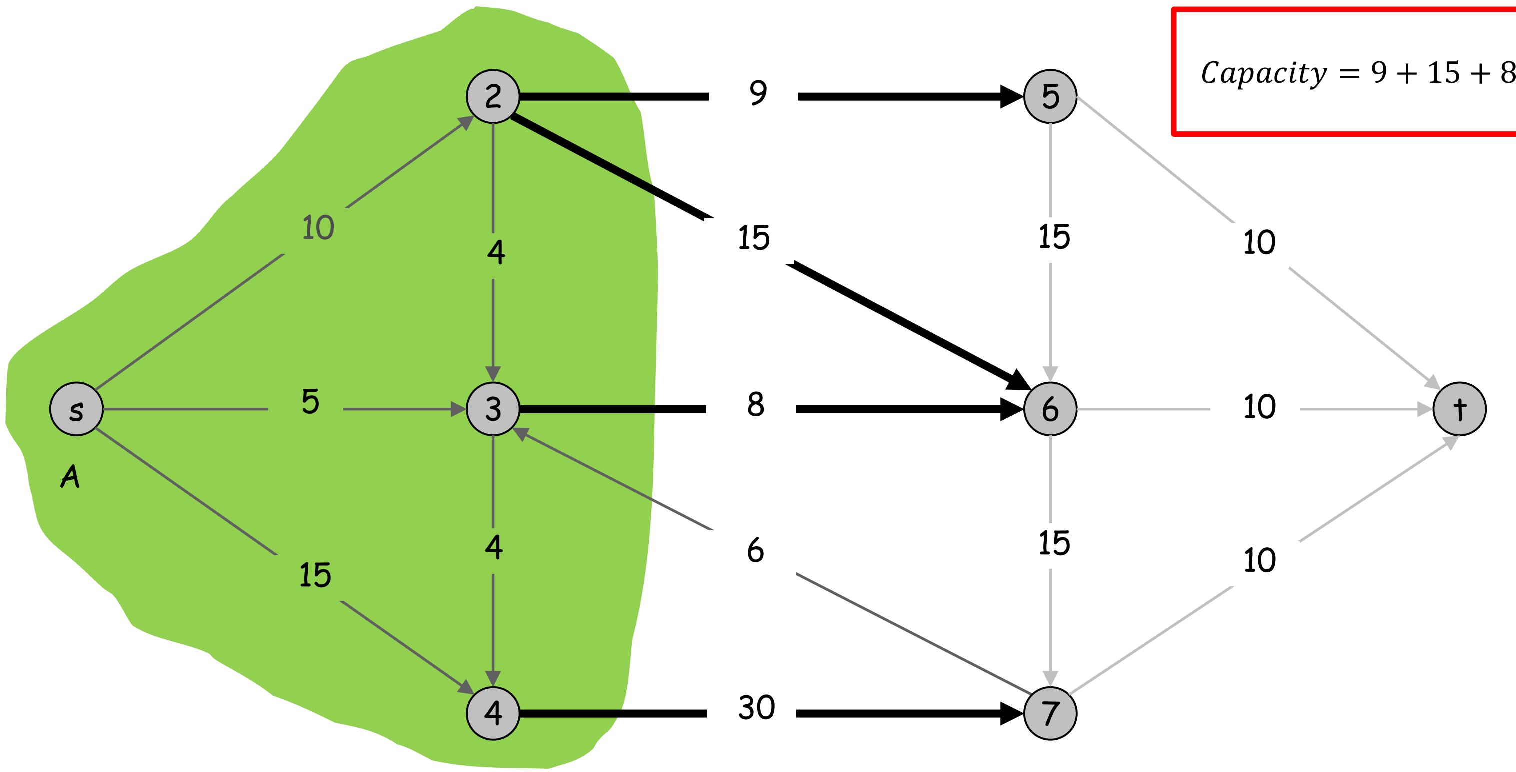
Def. The capacity of a cut (A, B): $cap(A, B) = \sum_{(u,v):u \in A, v \in B} c(u, v)$



s-t 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): $cap(A,B) = \sum_{(u,v):u\in A, v\in B} c(u,v)$



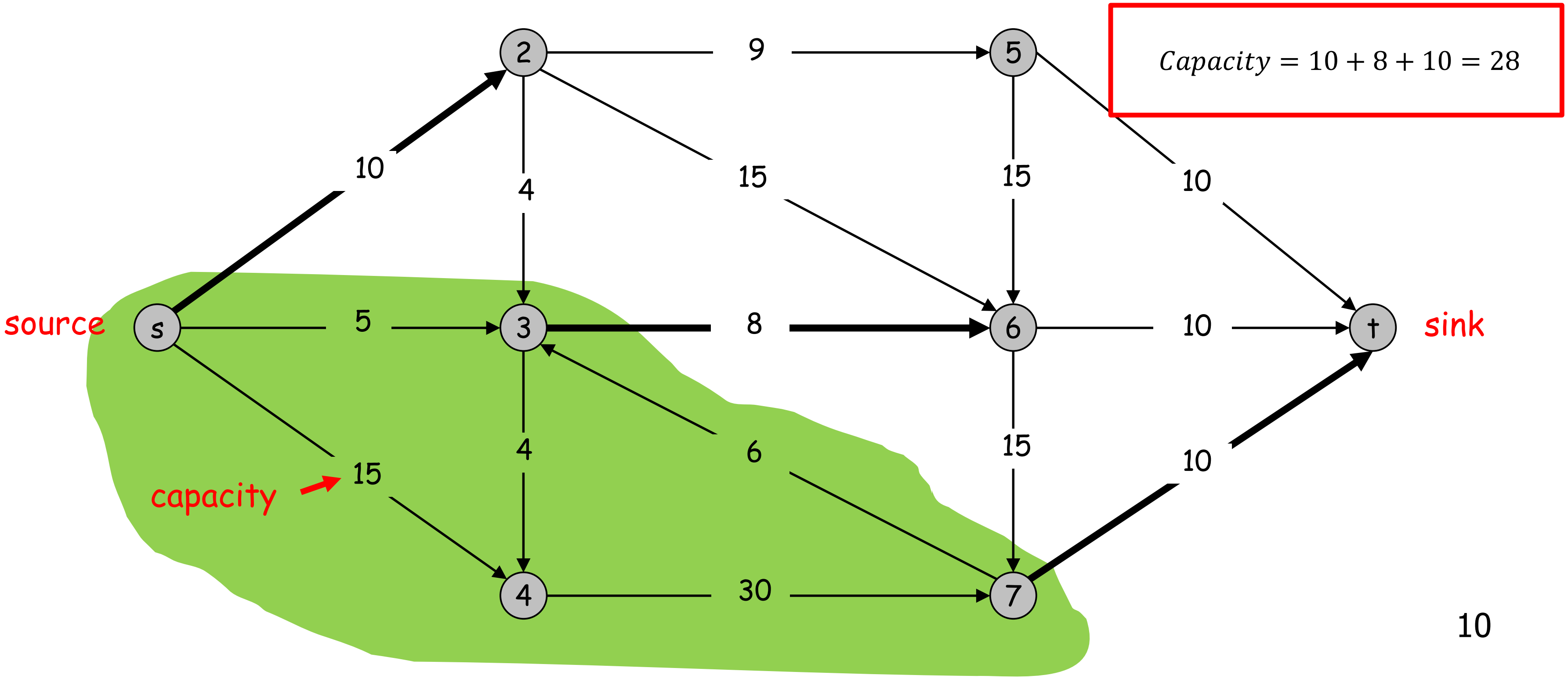
$Capacity = 9 + 15 + 8 + 30 = 62$

Minimum s-t Cut Problem

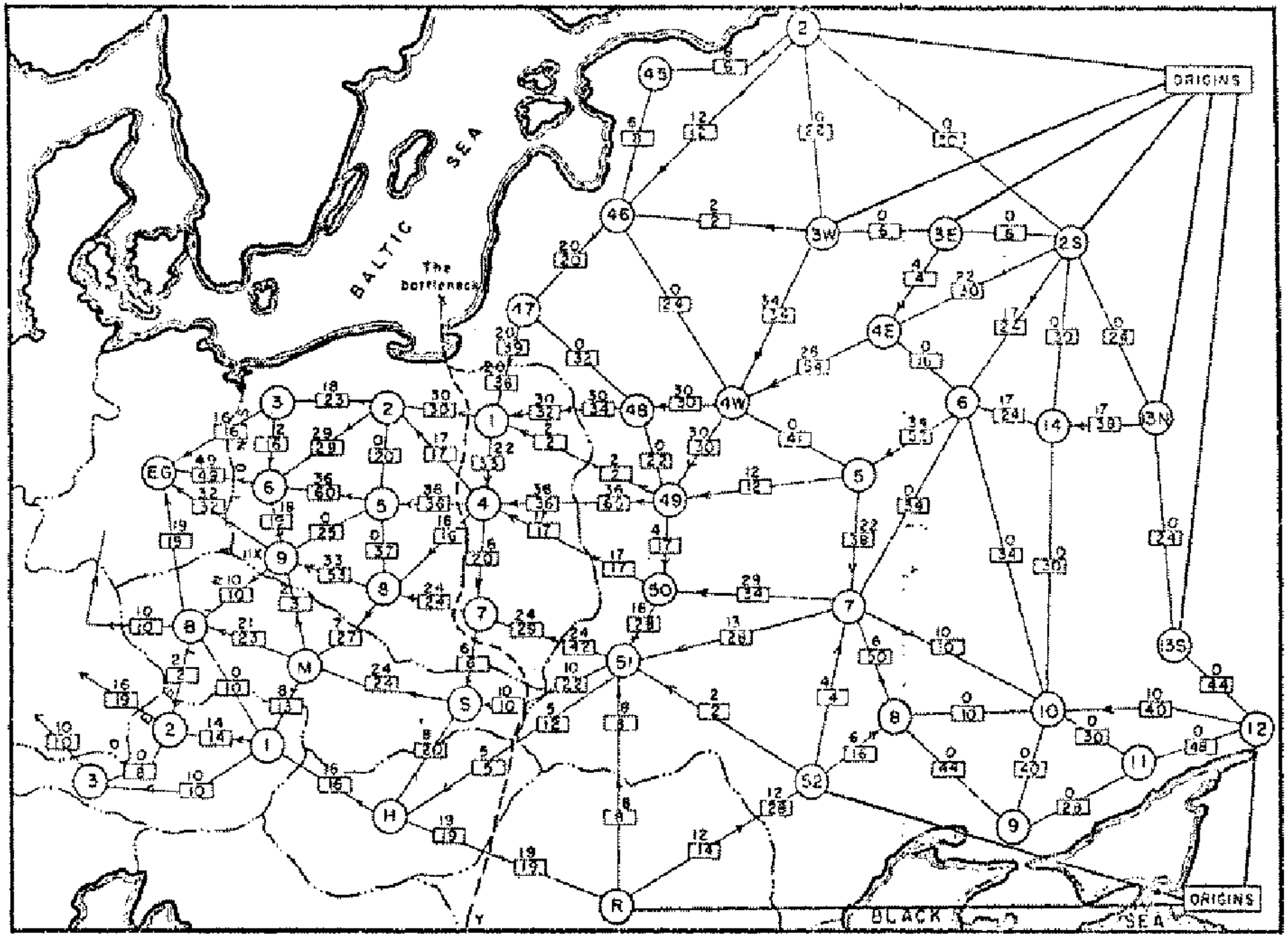
Given a directed graph G = (V, E) = directed graph and two distinguished nodes: s = source, t = sink.

Suppose each directed edge e has a nonnegative capacity $c(e)$

Goal: Find a s-t cut of minimum capacity



Soviet Rail Network



“Unclassified” on May 21, 1999.

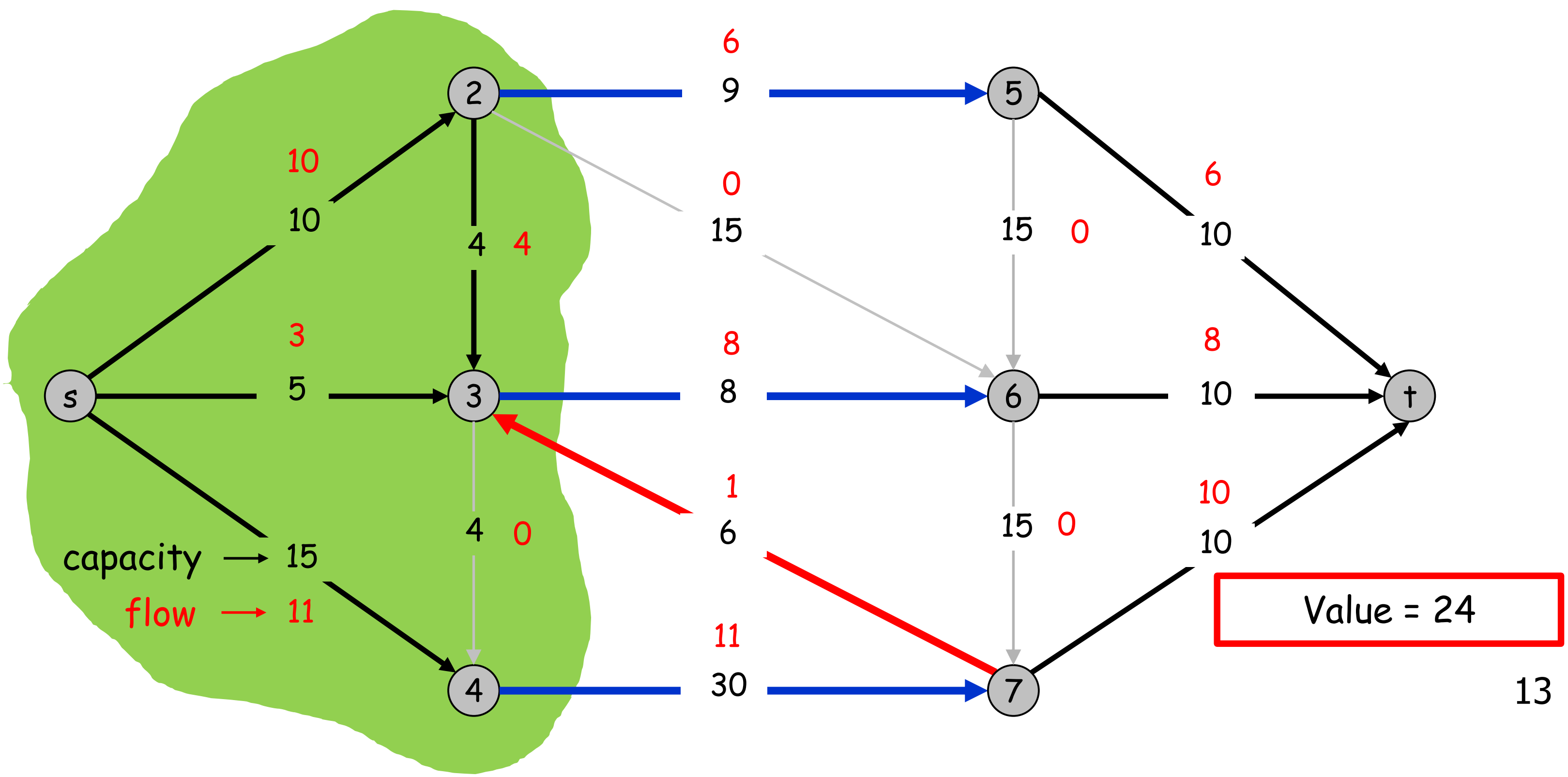
Outline

- Discuss how to prove a flow is optimal
  via min s-t cut problem
- Relate two problems
- Prove correctness of augmenting path
- Some exercise

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.

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



Proof of Flow value Lemma

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.

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

Proof.

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

By conservation of flow, all terms except v=s are 0 →

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

All contributions due to internal edges cancel out →

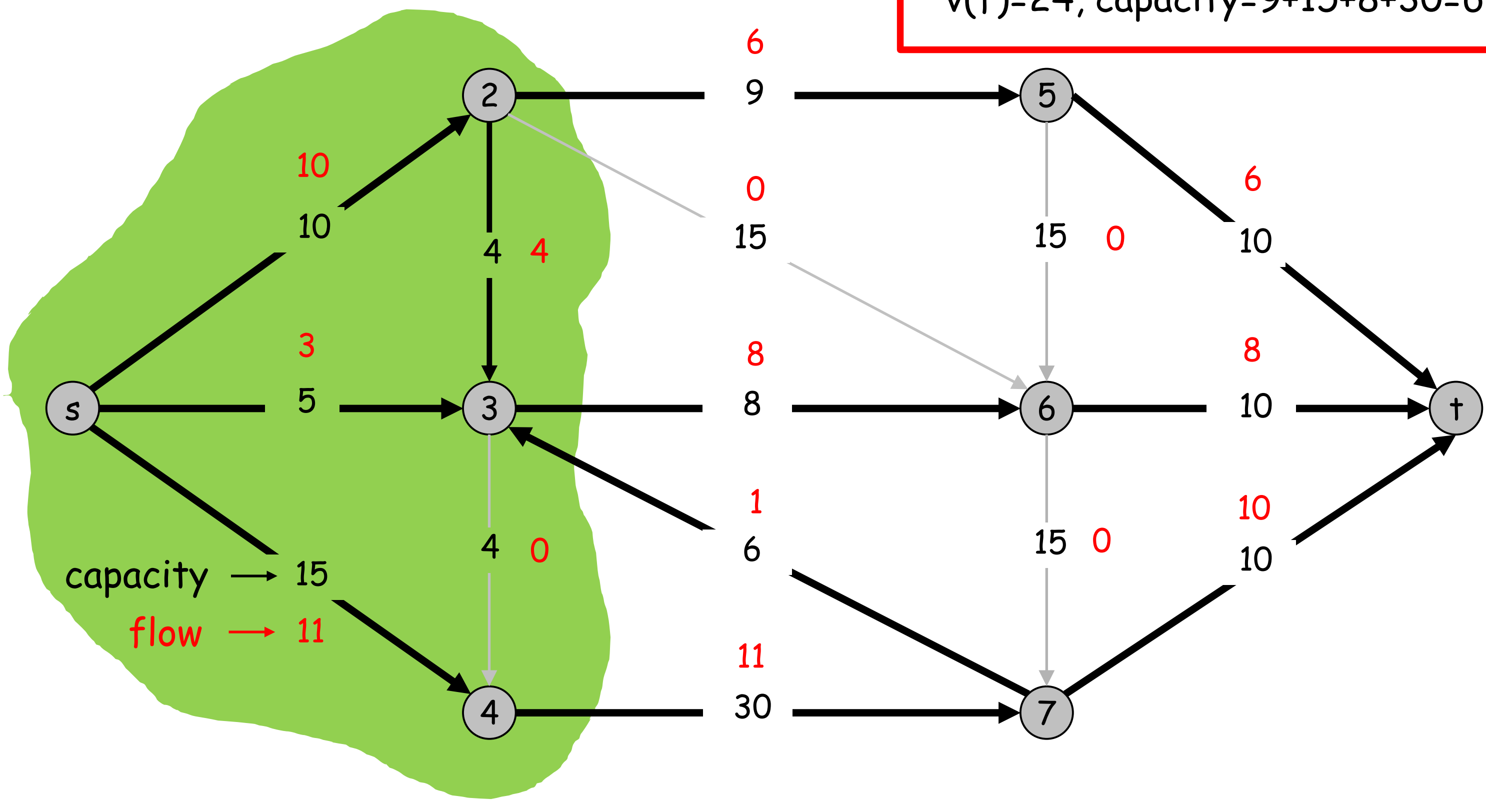
$$= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e)$$

Weak Duality of Flows and Cuts

Weak Duality. Let f be any flow, and let (A, B) be any s-t cut. Then the value of the flow is at most the capacity of the cut.

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

v(f)=24, capacity=9+15+8+30=62



Weak Duality of Flows and Cuts

Weak Duality. Let f be any flow, and let (A, B) be any s-t cut. Then the value of the flow is at most the capacity of the cut.

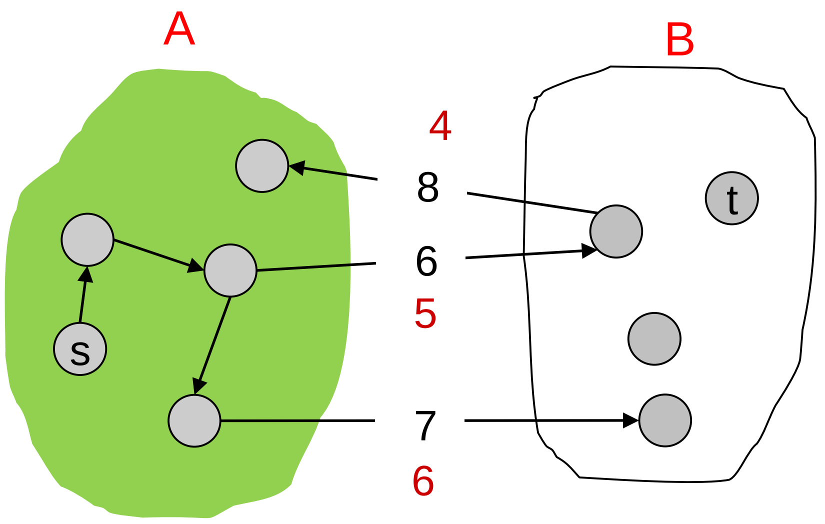
$$v(f) \le cap(A, B)$$

Proof.

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

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

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



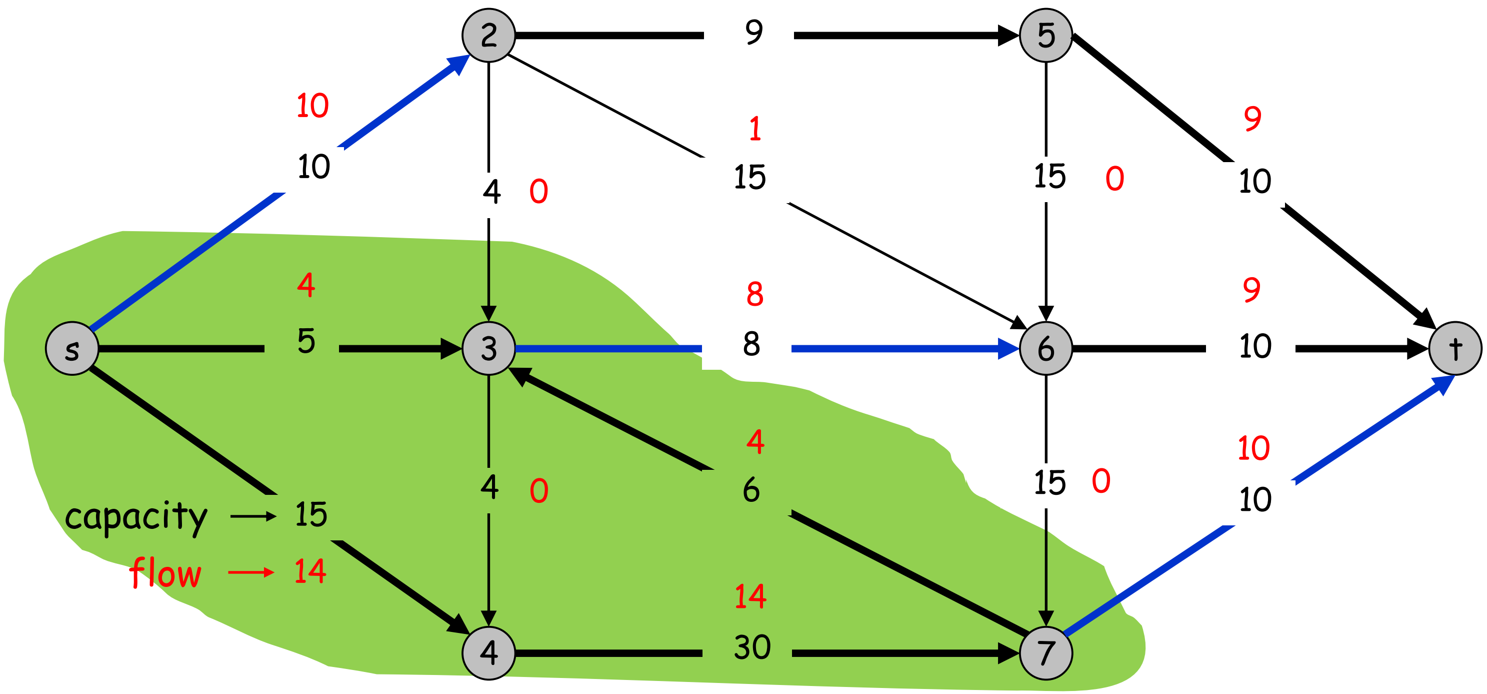
Certificate of Optimality

Corollary: Suppose there is a s-t cut (A,B) such that

$$v(f) = cap(A, B)$$

Then, f is a maximum flow and (A,B) is a minimum cut.

v(f)=28, cap(A,B)=28



Outline

- Discuss how to prove a flow is optimal
  via min s-t cut problem
- Relate two problems
- Prove correctness of augmenting path
- Some exercise

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 s-t flow is equal to the value of the min s-t cut.

Proof strategy. We prove both simultaneously by showing the 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 that path.

Pf of Max Flow Min Cut Theorem

(iii) => (i)

No augmenting path for f => there is a cut (A,B): v(f)=cap(A,B)

- 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, s $\in$ A.
- By definition of f, t $\notin$ A.

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

Running Time

Assumption. All capacities are integers between 1 and C.

Invariant. Every flow value $f(e)$ and every residual capacities $c_f(e)$ remains an integer throughout the algorithm.

Theorem. The algorithm terminates in at most $v(f^*) \leq nC$ iterations, if f^* is optimal flow.

Pf. Each augmentation increase value by at least 1.

Corollary. If C = 1, Ford-Fulkerson runs in O(mn) time.

Integrality theorem. If all capacities are integers, then there exists a max flow f for which every flow value f(e) is an integer.

Pf. Since algorithm terminates, theorem follows from invariant.

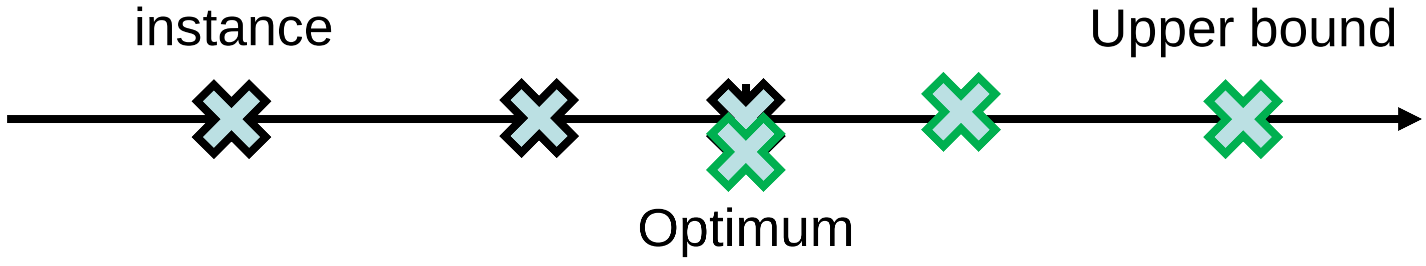
Summary

Max-flow min-cut theorem. The value of the max s-t flow is equal to the value of the min s-t cut.

Many optimization problems has two versions:

- Maximum version (finding the instance)
- Minimum version (finding the upper bound)

One problem is primal and one problem is dual.



Outline

- Discuss how to prove a flow is optimal
  via min s-t cut problem
- Relate two problems
- Prove correctness of augmenting path
- Some exercise

Exercise 1: without flow conservation

Suppose we are given a directed graph of water pipelines $G = (V, E)$ with a source s. The source vertex can produce up to C gallons of water every day (you get to choose the amount of production). Each edge e has a capacity of $c(e)$ gallons of water. At each node v there exists a tank that can store $s(v)$ gallons of water (for future use). On day t node v has a demand of $d[v, t]$ gallons of water. On some days we have a surplus of production and some other days we have too much demand so we have to use the stored water. We are running the system for T days. Design an algorithm that runs in polynomial time

Exercise 2: Residual Graph

One day, Sally comes up with a following linear time algorithm:

- For any directed graph with m edges and n vertices, it can compute a $s-t$ flow with flow value is at least ½ of maximum flow value .

Show how to use this to find exact maximum flow in $O(m \log F)$ time where F is the maximum flow value.

Exercise 3: Capacity

Recall that augmenting path takes $O(mF)$ time. Shows how to solve maximum flow in $O(mn \log F)$.