CSE 6140/ CX 4140:

Computational Science and Engineering ALGORITHMS

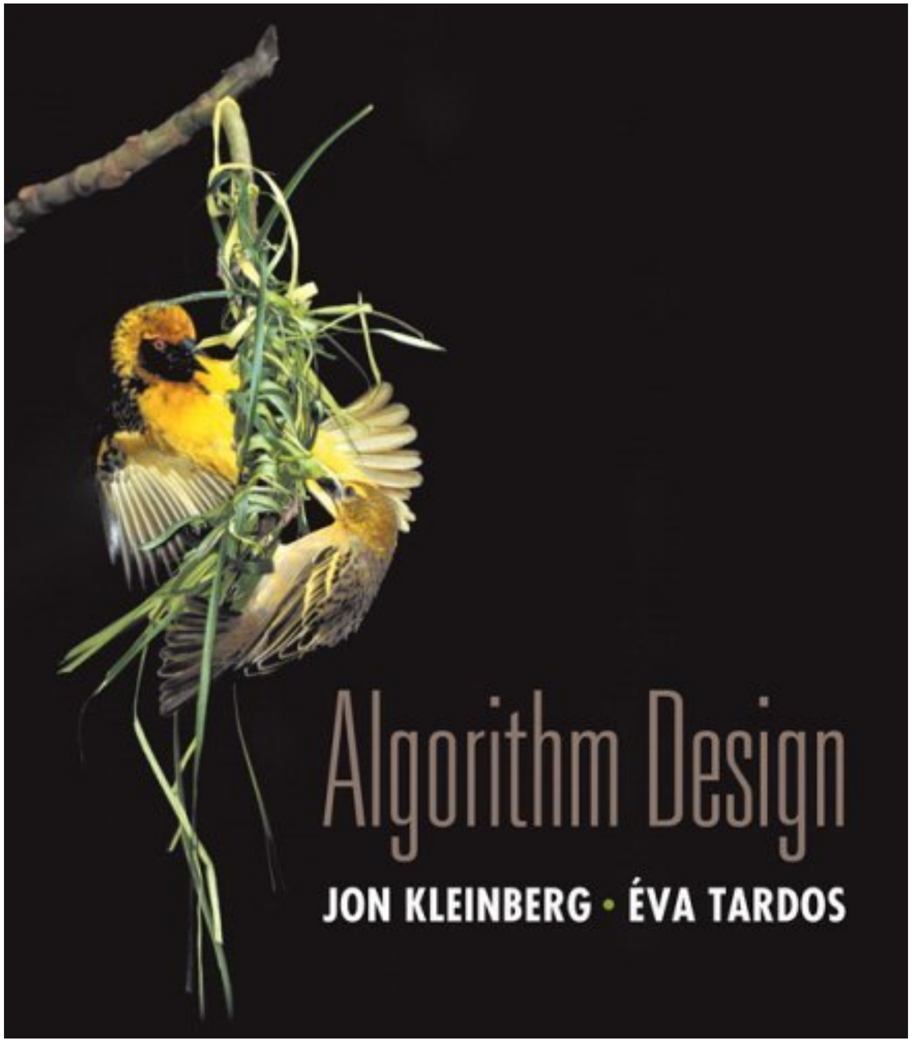
Instructor: Anne Benoit

Visiting Associate Professor, CSE

Based on slides by Bistra Dilkina

CLRS: Chapter 26 & KT: Chapter 7

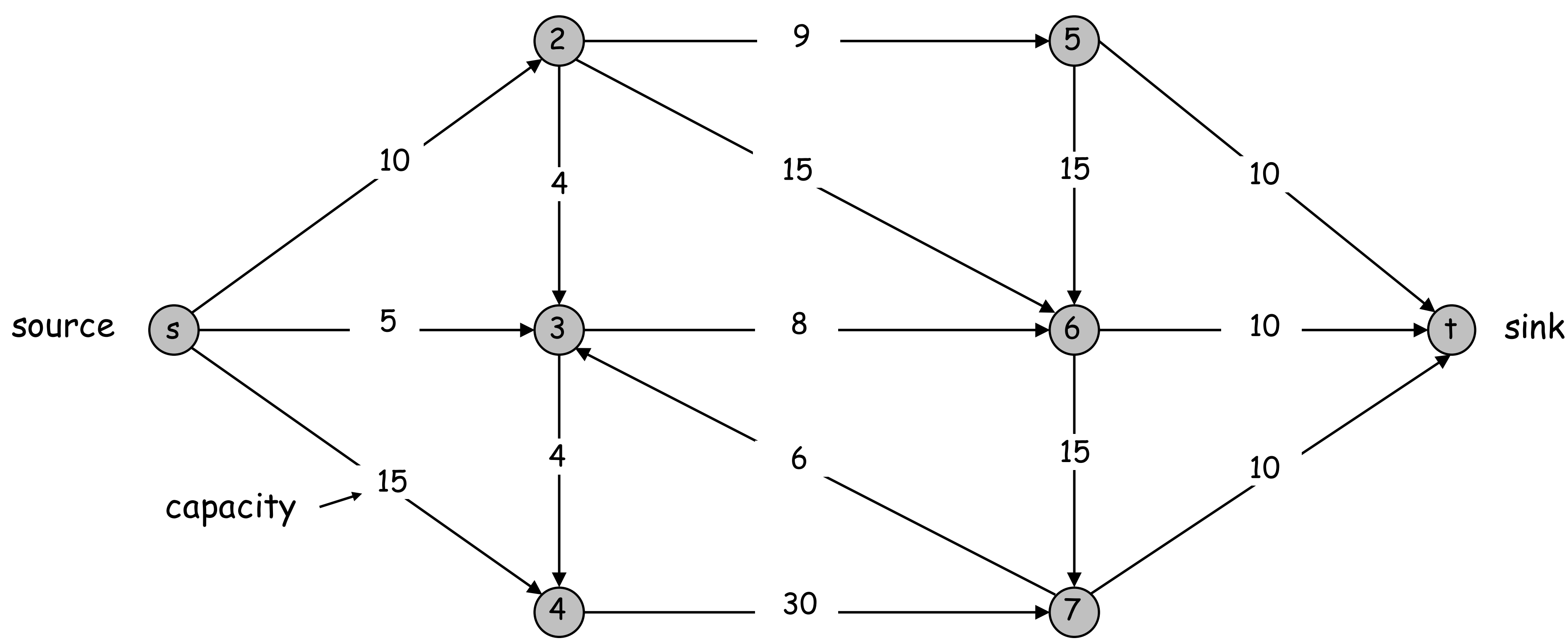
Network flows – Part 2



Flow Network

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.

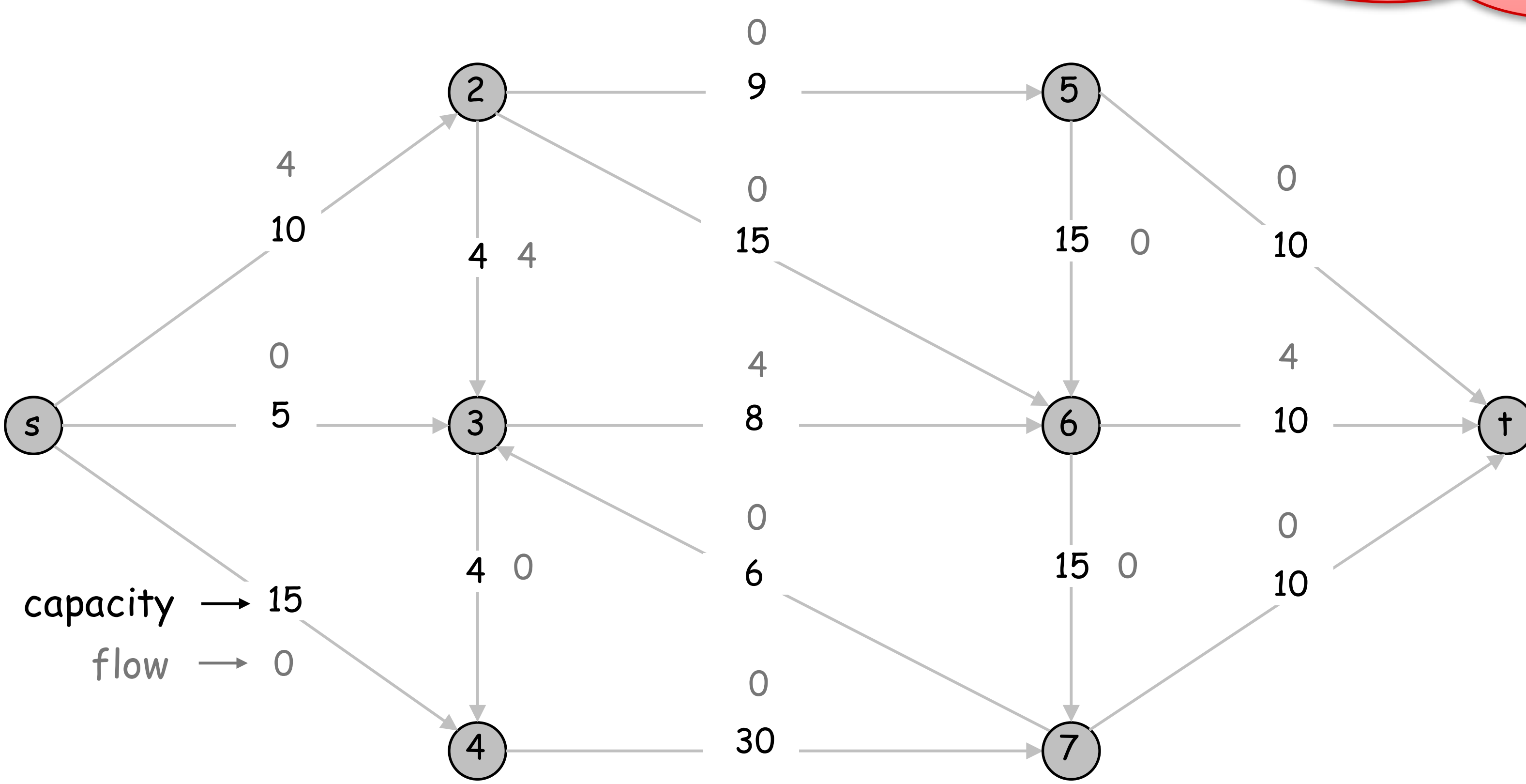


Flows

Def. An **s-t flow** is a function f from E to real numbers that satisfies:

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

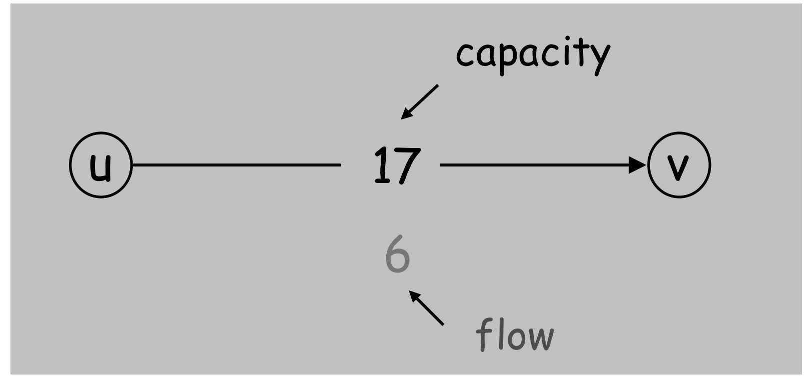
water flowing
from source to sink



Residual Graph

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

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



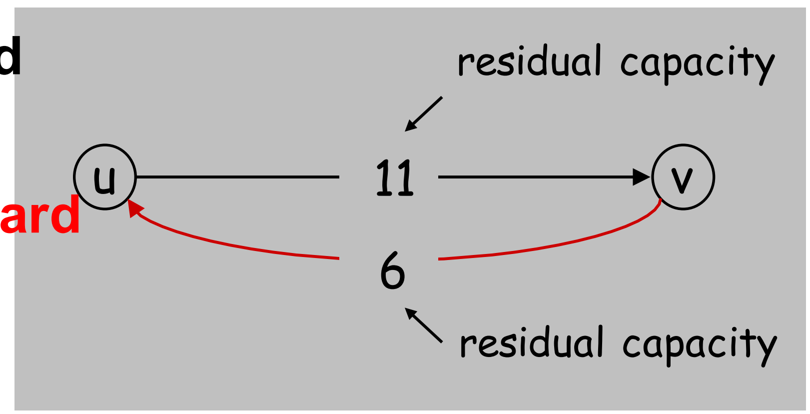
Residual edge.

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

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

forward

backward



- If e in E:
  - unused capacity
- If e^R in E:
  - ability of sent less water, or “undo” flow

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

- Residual edges with positive residual capacity: $E_f = \{e : c_f(e) > 0\}$.
- Edges in E with flow=capacity are only present in reverse direction in E_f
- Edges in E with no flow are only present in their original direction in E_f

Augmenting Path Algorithm

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

   while (there is an s-t path P in Gf) {
      f ← Augment(f, c, P)
      update Gf
   }
   return f
}
```

```
Augment(f, c, P) {
   b ← bottleneck(P)
   foreach e ∈ P {
      if (e ∈ E) f(e) ← f(e) + b
      else [eR ∈ E] f(eR) ← f(eR) - b
   }
   return f
}
```

Min residual capacity of an edge in P

forward edge

backward edge

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. [Elias-Feinstein-Shannon 1956, Ford-Fulkerson 1956]
The value of the max flow is equal to the value of the min s-t cut.
We have the equivalence between:

(i) There exists an s-t 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.

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. [Elias-Feinstein-Shannon 1956, Ford-Fulkerson 1956]
The value of the max flow is equal to the value of the min s-t cut.
We have the equivalence between:

(i) There exists an s-t 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.

Proof:
(i) $\Rightarrow$ (ii) This was the corollary to weak duality lemma: $v(f) \leq cap(A,B)$ for all s-t cut (A,B). Since $v(f)=cap(A,B)$, f is a max flow.

(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.

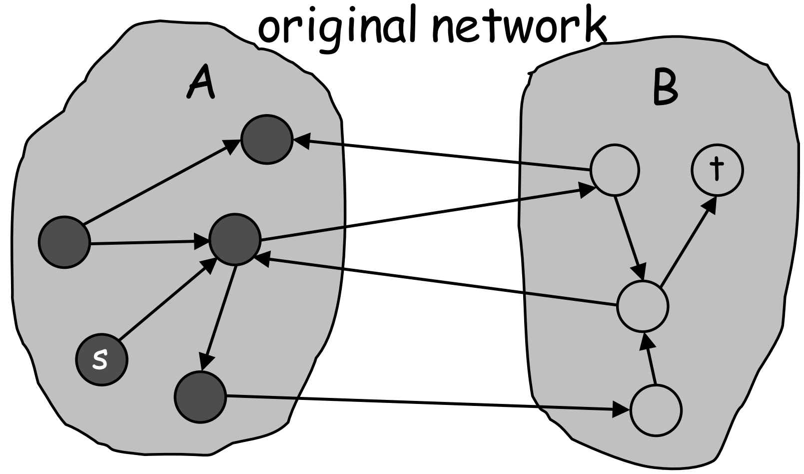
Proof of Max-Flow Min-Cut Theorem

(iii)$\Rightarrow$ (i)   (iii)There is no augmenting path relative to f.
(i)There exists an s-t cut (A, B) such that 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 G_f.
- By definition of A, s $\in$ A.
- By definition of f (no augmenting path), t $\notin$ A.
- Observation: No edges of the residual graph go from A to B.

- **Claim 1**: If e$\in$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$\in$E goes from B to A, then f(e)=0.
- Proof: Otherwise residual edge would go from A to B.

original network

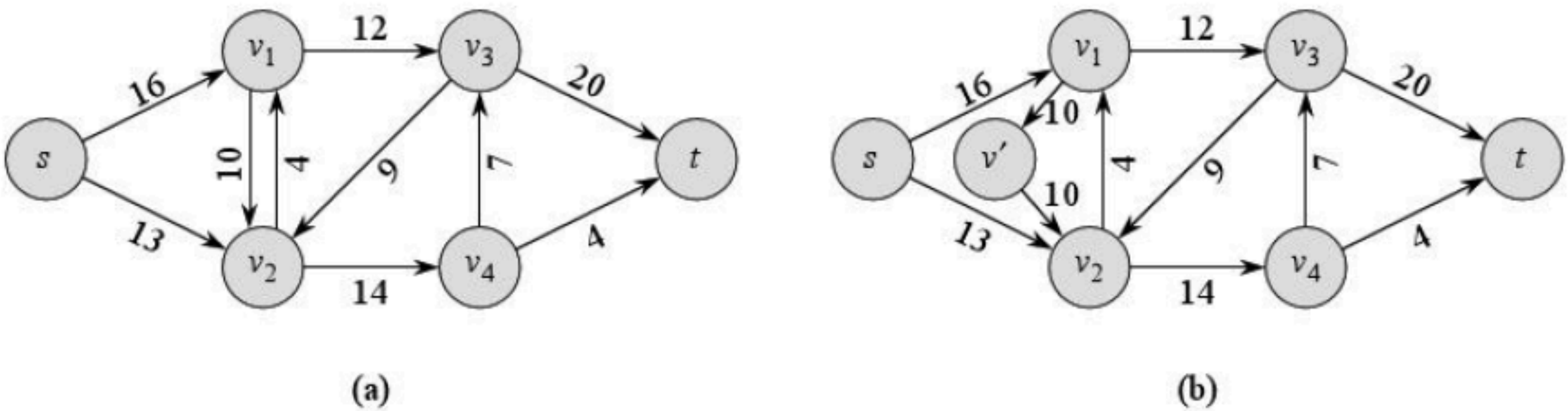


$$\begin{aligned} v(f) &= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) \\ &= \sum_{e \text{ out of } A} c(e) \\ &= cap(A,B) \quad \blacksquare \end{aligned}$$

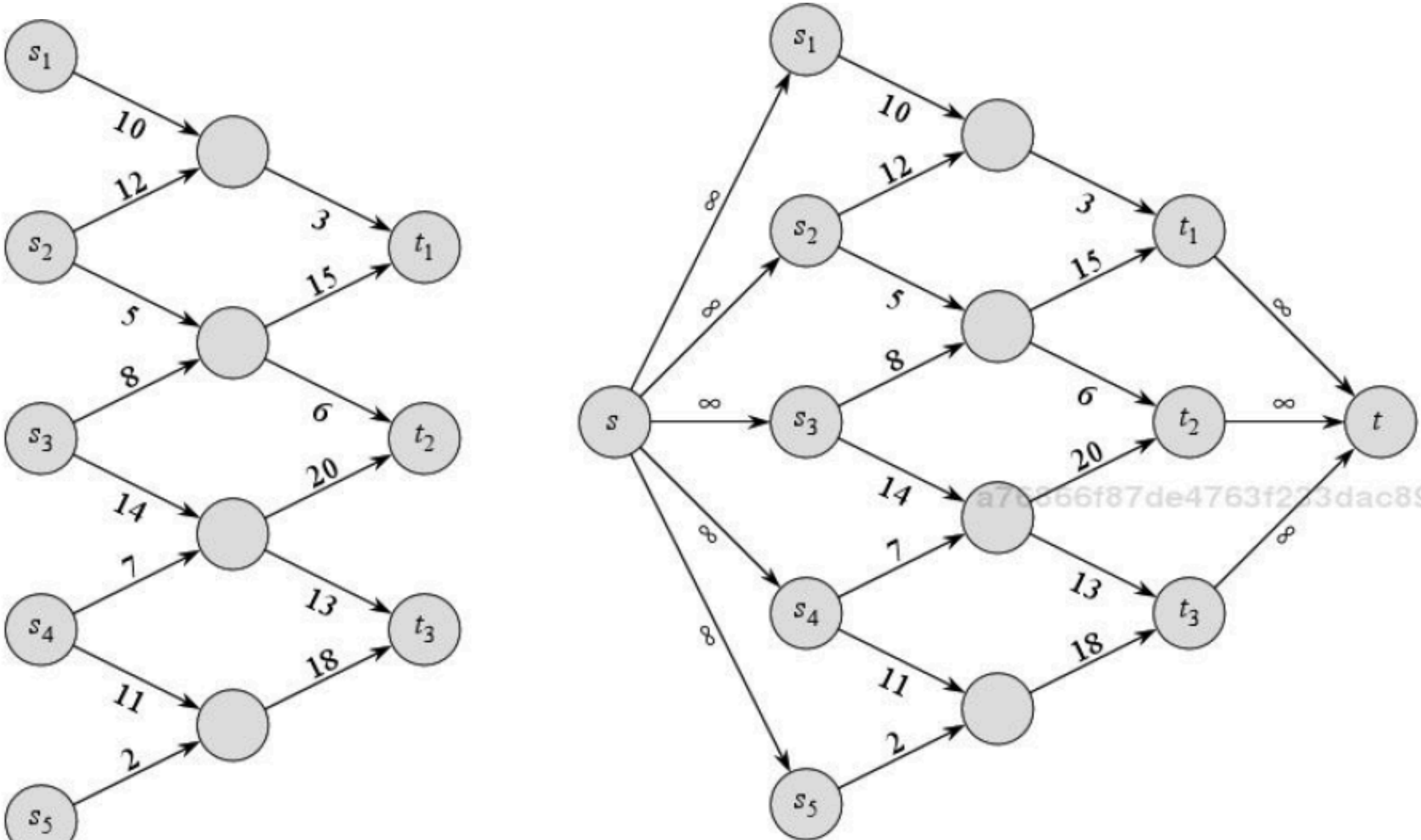
What comes next

- We have shown at termination, Ford-Fulkerson is optimal
- Need to show feasibility is preserved at every iteration of Ford-Fulkerson
- Need to show termination
- Need to analyze space and time
- First, let's look at some of our simplifying assumptions
  - Single source and target – not true for our railroad example
  - No edges in both directions between a pair of vertices

Edges in both directions (reduction)



Multiple sources and sinks (reduction)



(a) (b)

Correctness: Augmenting the flow preserves feasibility

Given a flow network G and flow f, and the corresponding residual graph G_f with capacities c_f, then flow f' produced after adding an augmenting path P is feasible in G.

Proof. We must show that capacity and flow conservation constraints hold.

1) f' differs from f only on edges in P, so we only need check those capacities

- If edge e=(u,v) in P is forward edge (i.e. in E) then

$$f'(e) = f(e) + \text{bottleneck}(P,f) \leq f(e) + c_f(e) \leq f(e) + (c_e - f(e)) = c_e$$

- If edge (u,v) in P is backward edge (i.e. e=(v,u) is in E)

$$c_e \geq f(e) \geq f'(e) = f(e) - bottleneck(P,f) \geq f(e) - c_f(e) = f(e) - f(e) = 0$$

smallest residual capacity

2) Need to check conservation of flow at each internal node on P

Say node v with edges (u,v) and (v,w) in P

- It is easy to check that since f satisfies conservation and the path pushed equal flow on (u,v) and (v,w) the conservation is preserved
- Four cases depending of whether (u,v) and (v,w) are forward or backward edges

Termination

We will show FF always terminates (in an integral flow) - given integer capacities in input

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

Proof: (by induction)
1) True in iteration 0 when all flows are 0

2) Assume true in iteration k

3) Since all residual capacities are integer at iter k, then at iteration k+1 the augmenting path has bottleneck(P) also integer (equal to smallest cap.)
Thus (k+1)-flow f'=f+bottlneck(P,f) is sum of 2 integers and hence integer
By definition of residual capacities, if original capacities and flow f' are integer then the new residual capacities at iteration k+1 are also integer

Termination

We will show FF always terminates (in an integral flow)

Claim. Given a flow f and augmenting path P, then resulting new flow f' has value v(f') greater than v(f)

Proof:
By construction the first edge in P is out of s in residual graph G_f

P is a simple path so no other edge in P touches s

Original graph G has no incoming edges in s, so the first edge in P coming out s has to be forward edge in residual graph

Hence we increase the flow on this edge by bottleneck(P,f) > 0, and change no other edge of s

=> v(f')=v(f) + bottleneck(P,f) > v(f)

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

Termination

Assumption. All capacities are integers between 1 and C.
Intergality Invariant. Every flow value $f(e)$ and every residual capacity $c_f(e)$ remains an integer throughout the algorithm.

Theorem. The algorithm terminates in at most $v(f^*) \le nC$ loop iterations.
Pf. To get a bound on maxflow: the maximum flow cannot be greater than the capacity leaving the source, so even if source is connected to all other nodes with capacity C, maxflow cannot exceed nC.

Each augmentation increases value by at least 1 so at most maxflow $v(f^*)$ iterations (from our Integrality Invariant always integer and Claim always increasing), hence at most nC augmenting paths/loops.

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 FF algorithm terminates and finds an optimal solution, theorem follows from integrality invariant.

Running Time

Running time of Ford-Fulkerson: $O(mnC)$. Space: $O(m+n)$.

Proof:

$O(nC)$ iterations from before. Let original graph have n nodes and m edges.
At each iteration we need to construct residual graph and find a path.
Residual graph has at most $2m$ edges.
-Finding a path is $O(m+n)$ using BFS or DFS.
We assume every node has at least one edge, so $O(m+n)=O(m)$.
-Changing the flow with augmenting path is $O(n)$ since P is a simple path with at most $n-1$ edges.
-Space: we use adjacency lists: $O(m+n)$.

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

Integer Capacities assumption

- Critical to show that FF terminates

- But can work with rational capacities (ratio of 2 integers), if we simple scale up all capacities by least common multiple

- With real capacities, we can forever augment flow by small fraction

Note: The min-cut max-flow equivalence, however, holds in general for any capacities

Applications of Maxflow when C=1

Maximum bipartite matching

- Reducing MBM to max-flow

Edge-disjoint paths

- another reduction

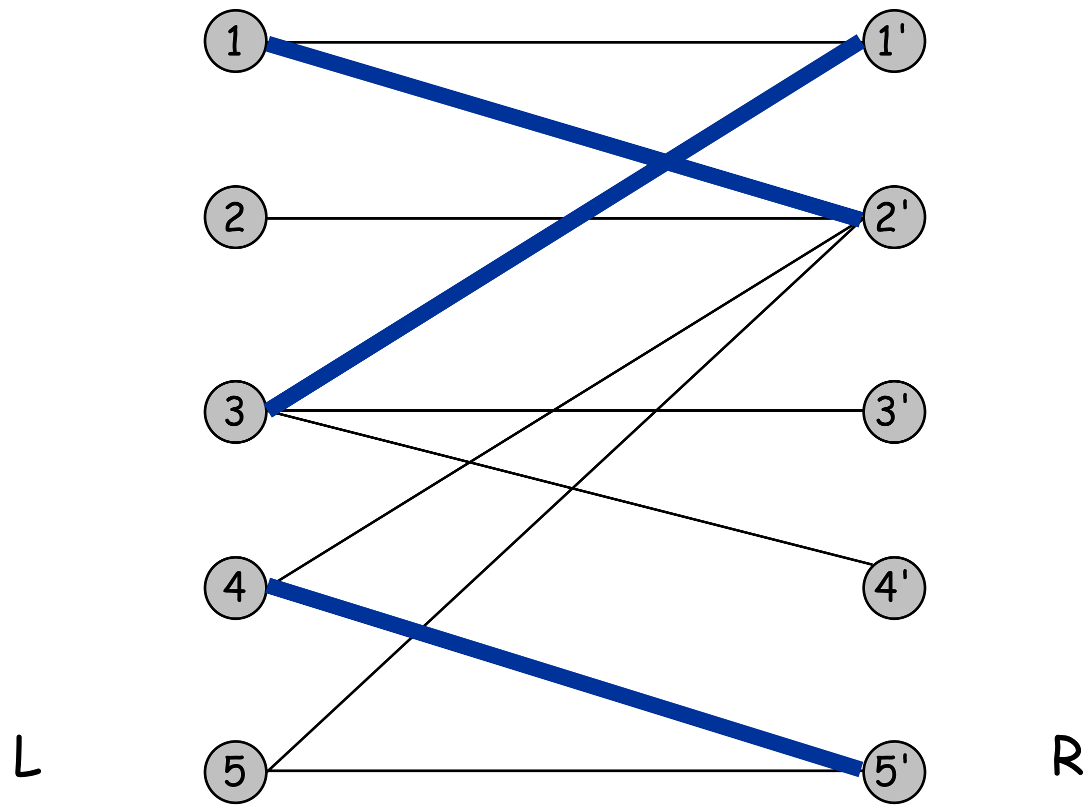
KT 7.5

Bipartite Matching

Bipartite Matching

Bipartite matching.

- Input: undirected, bipartite graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a matching if each node appears in at most 1 edge in M.
- Max matching: find a max cardinality matching.



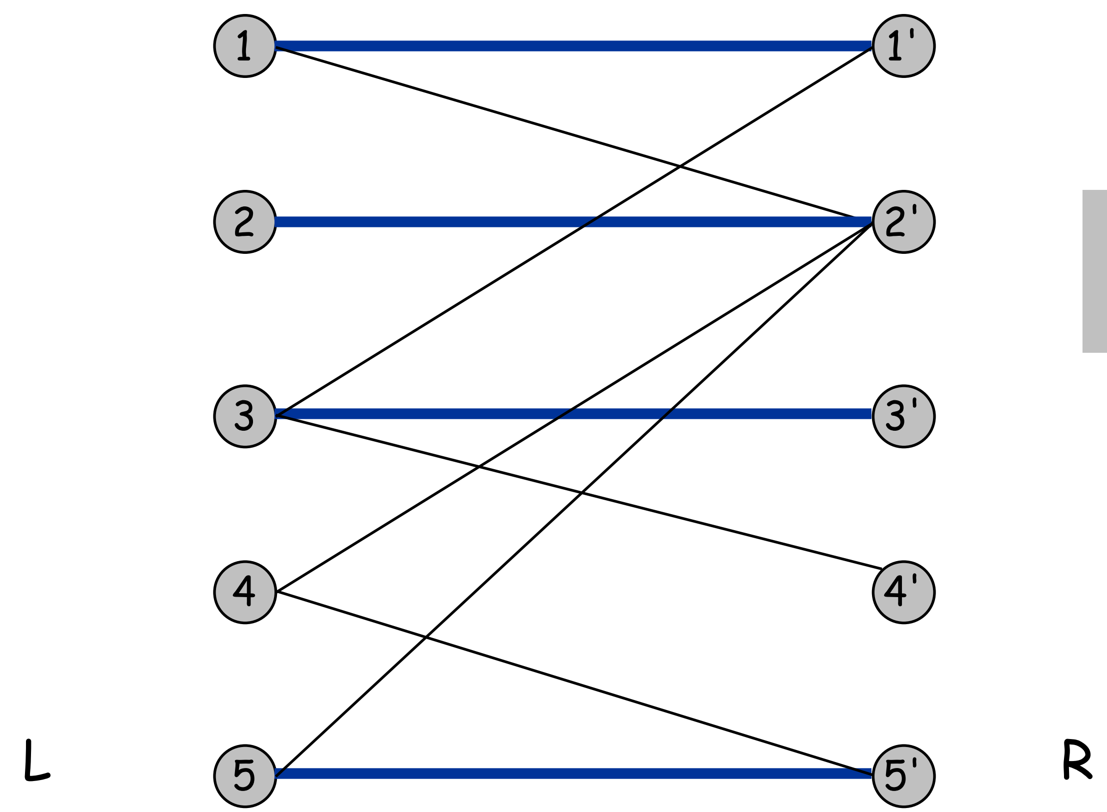
matching

1-2', 3-1', 4-5'

Bipartite Matching

Bipartite matching.

- Input: undirected, bipartite graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a matching if each node appears in at most 1 edge in M.
- Max matching: find a max cardinality matching.



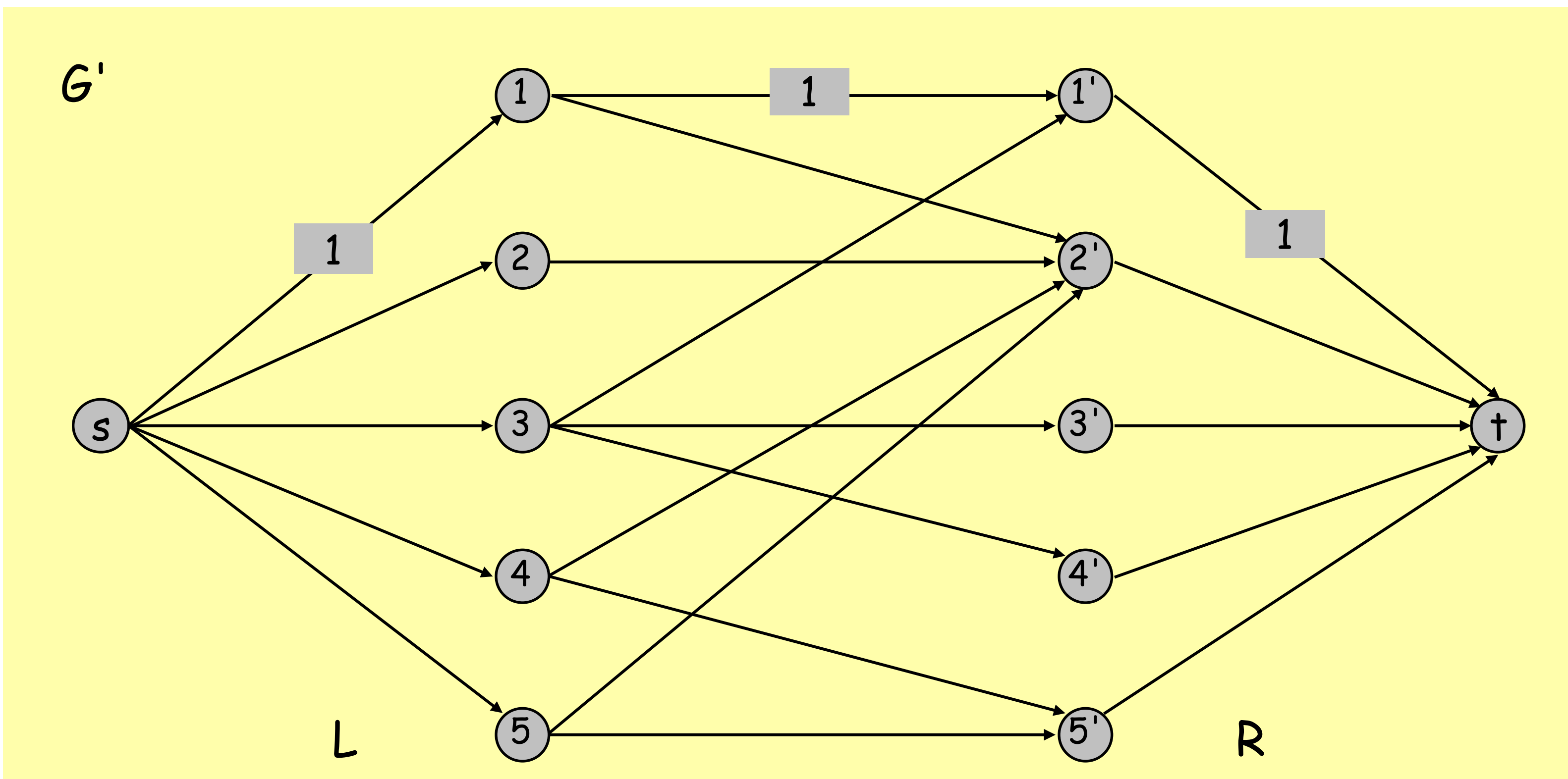
<u>max matching</u>

1-1', 2-2', 3-3' 4-4'

Reducing Bipartite Matching to Maximum Flow

Reduction to Max flow.

- Create directed graph $G' = (L \cup R \cup \{s, t\},\ E')$.
- Direct all edges from L to R, and assign capacity 1.
- Add source s, and capacity 1 edges from s to each node in L.
- Add sink t, and capacity 1 edges from each node in R to t.



Bipartite Matching: Proof of Correctness

Theorem. Max cardinality matching in G = value of max flow in G'.

Proof: We need two statements

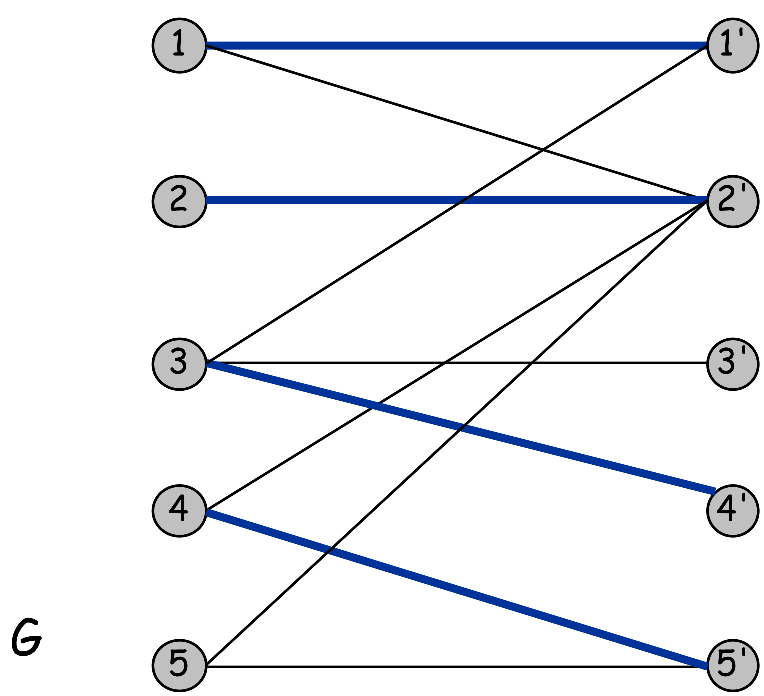
- max. matching in $G \leq$ max flow in G'
- max. matching in $G \geq$ max flow in G'

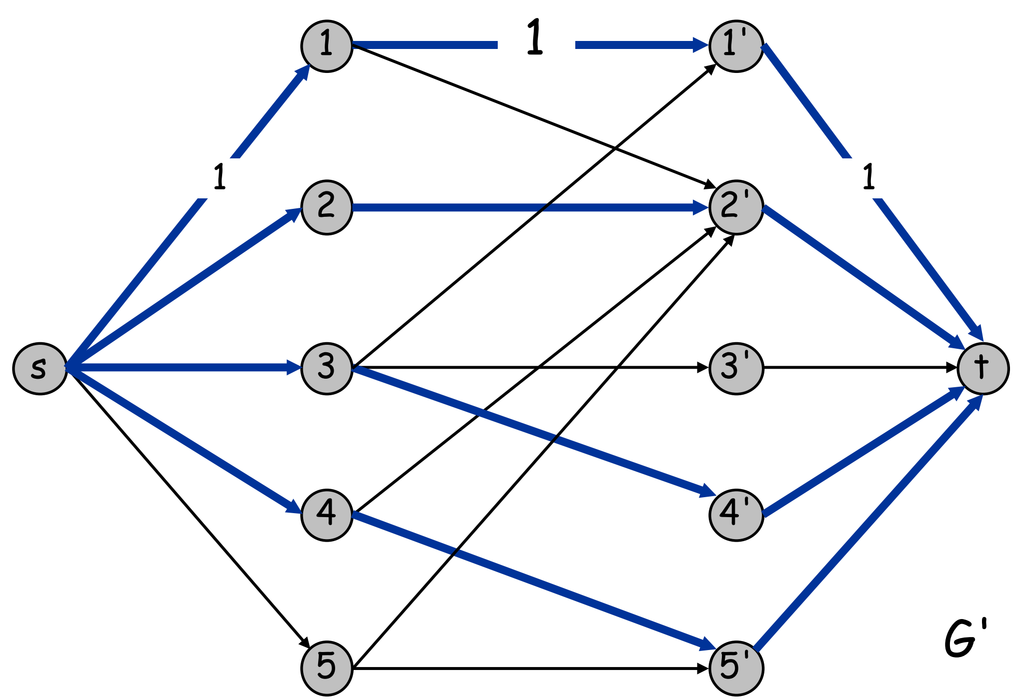
Bipartite Matching: Proof of Correctness

Theorem. Max cardinality matching in G = value of max flow in G'.

Pf. max. matching in $G \leq$ max flow in G'

- Given max matching M of cardinality k.
- Consider flow f that sends 1 unit along each of k paths (s,e,t for e in M)
- f is a feasible flow of value k (the cut ($L \cup s$, $R \cup t$) has flow of k).
- hence maxflow is at least as good as f
- i.e. maxflow >= k (= max matching)





Bipartite Matching: Proof of Correctness

Theorem. Max cardinality matching in G = value of max flow in G'.

Pf. max. matching in $G \geq$ max flow in G'

- Let f be a max flow in G' of value k.
- Integrality theorem $\Rightarrow$ f is integral; all capacities are 1 $\Rightarrow$ $f(e)$ is 0 or 1.
- Consider M = set of edges from L to R with $f(e) = 1$.
  - each node in L and R participates in at most one edge in M (due to s,t)
  - Size? consider cut $(L \cup s, R \cup t)$, flow across cut is k, hence $|M| = k$