Review....

Ford-Fulkerson algorithm for max-flow: repeatedly augment flow along paths in the residual graph

If capacities are integers, F-F finds an integer valued flow

Running time: $O((m+n)OPT)$, where OPT is the value of the max flow

Correctness?

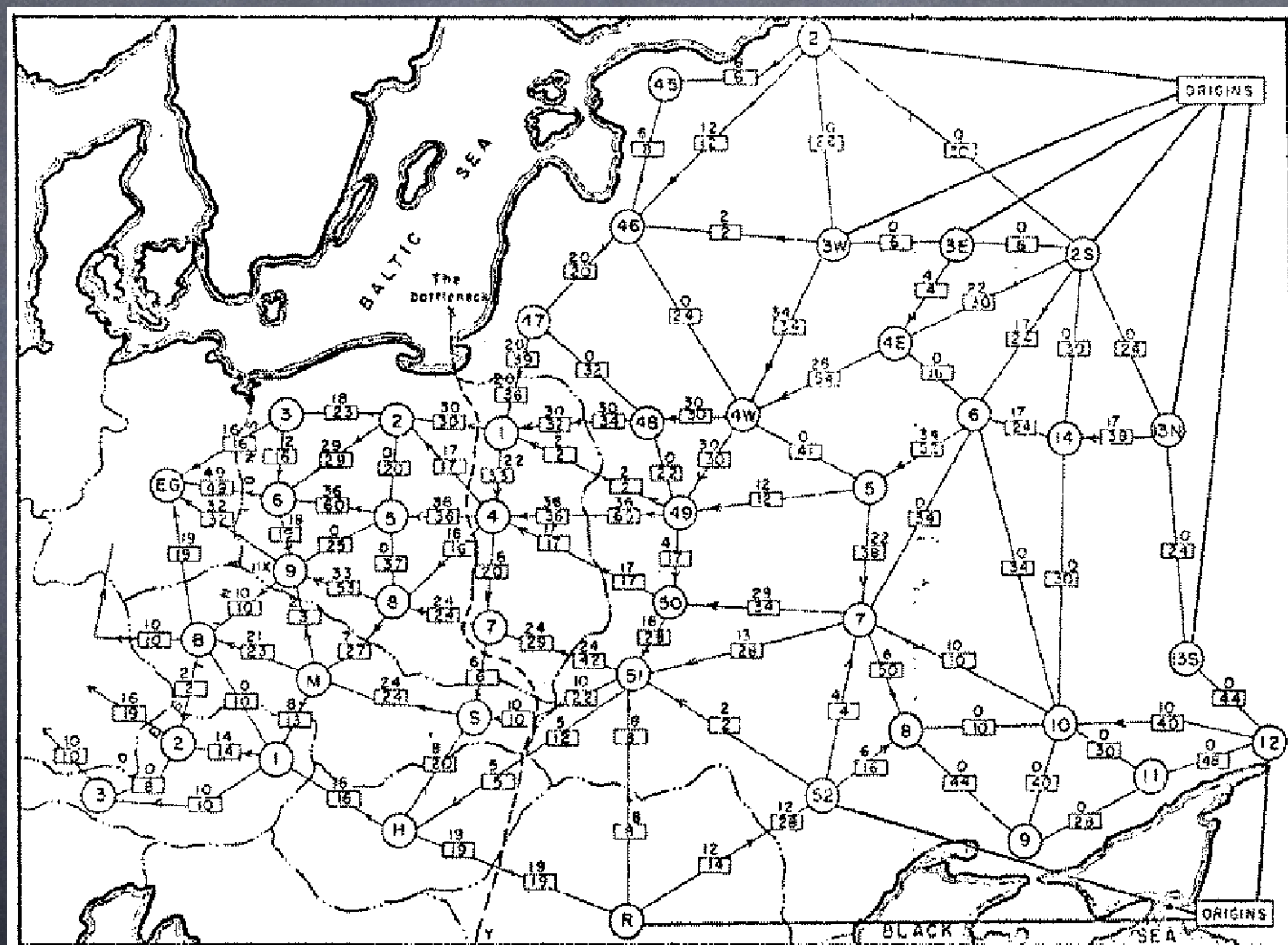
Today

1) Prove that Ford-Fulkerson algorithm finds a maximum flow by proving the Max-Flow Min-Cut Theorem

- Fundamental result in combinatorial optimization (best know example of duality)
- Independently proven in 1956 by Ford and Fulkerson AND by Elias, Feinstein and Claude Shannon

2) Bipartite matching

Flows and Cuts are Intimately Related!



Reference: On the history of the transportation and maximum flow problems.
Alexander Schrijver in Math Programming, 91: 3, 2002.

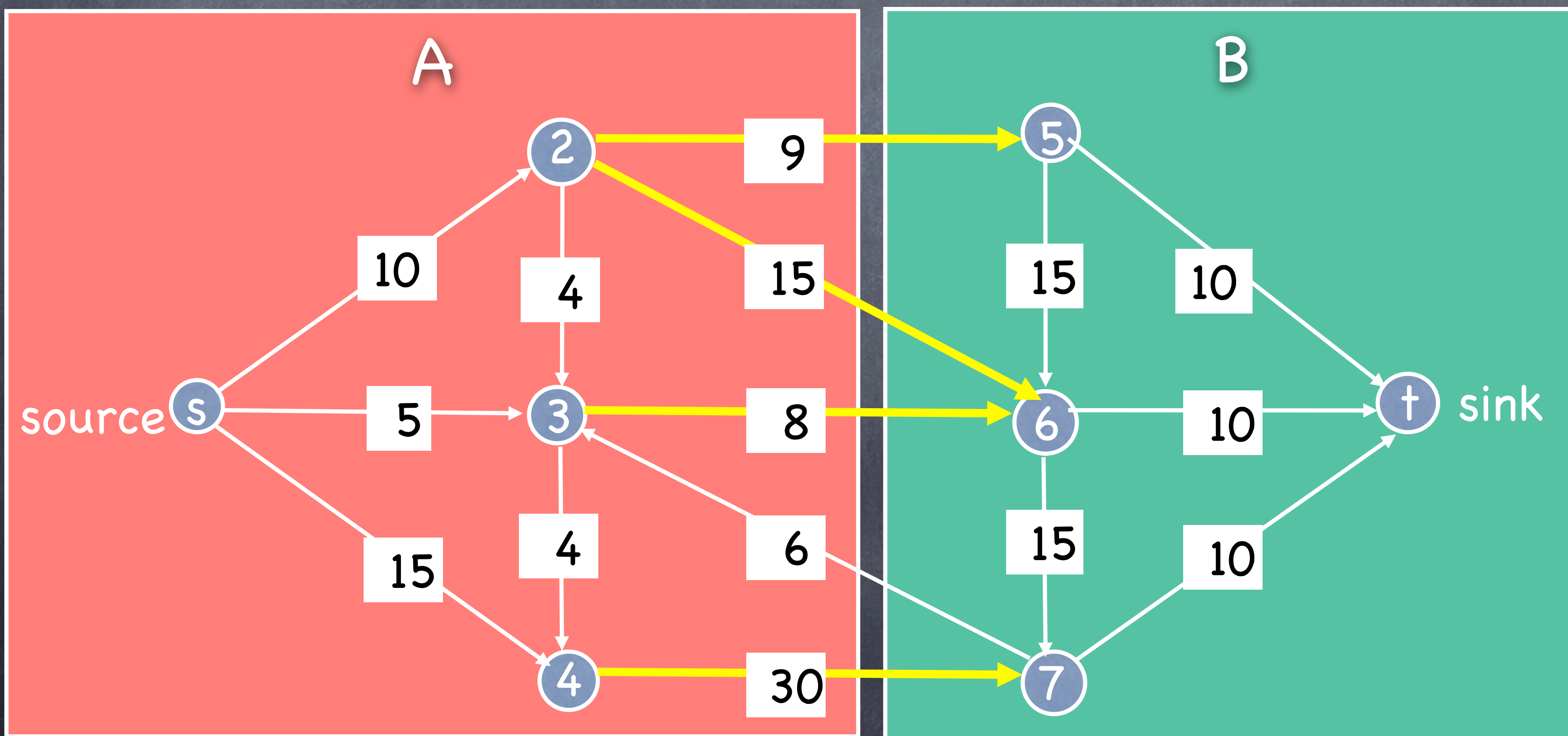
Outline

- Definitions: s-t cuts and their capacities
- Flow value lemma: how to measure a flow using different s-t cuts in the network
- The main event: Max-flow Min-Cut Theorem

Cuts

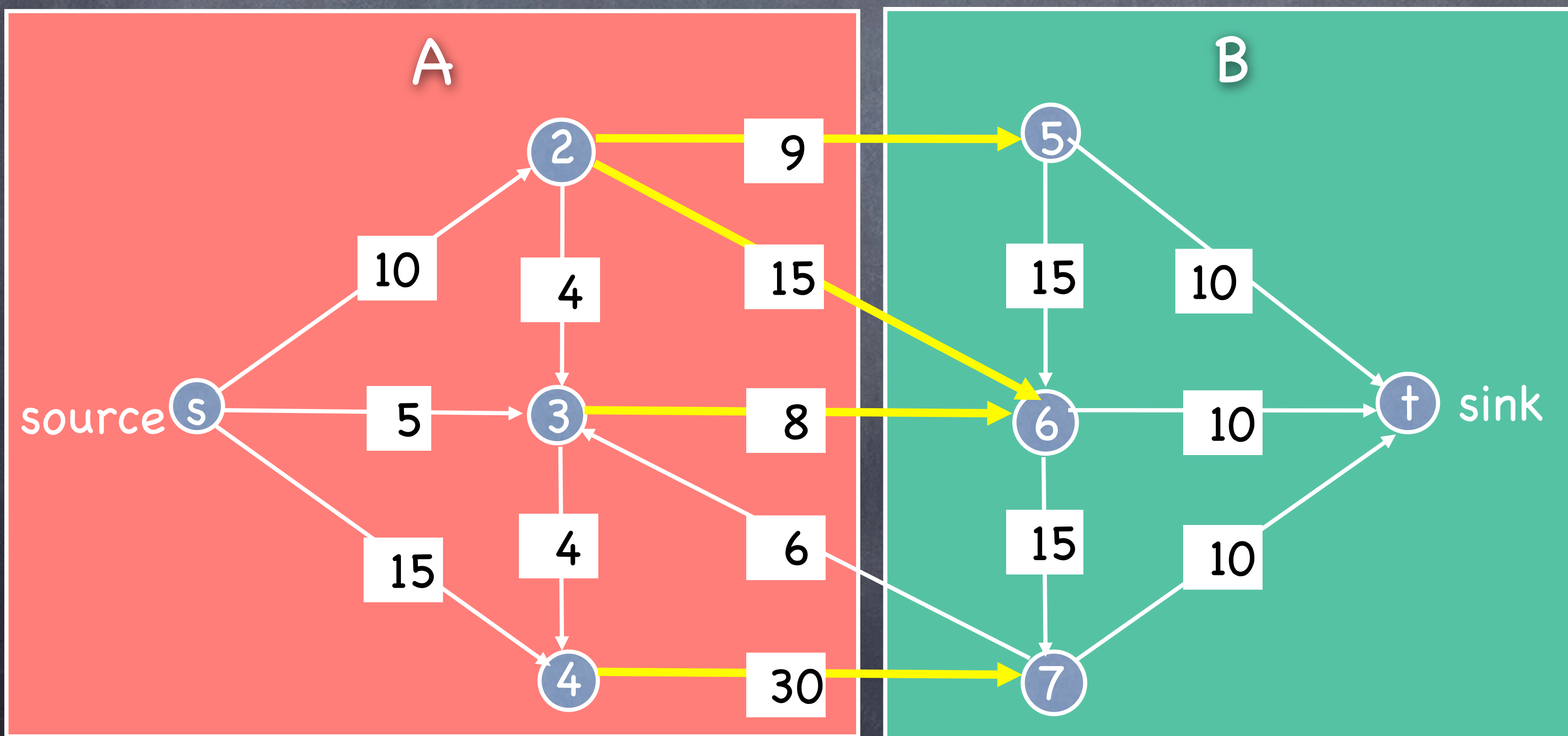
- An s-t cut is a partition (A, B) of V with $s \in A$ and $t \in B$.
- The capacity of a cut (A, B) is: $\sum_{e \text{ out of } A} c(e)$

capacity of A-B cut = 9 + 15 + 8 + 30 = 62



Cuts

capacity of cut = 9 + 15 + 8 + 30 = 62
(Capacity is sum of weights on edges leaving A.)

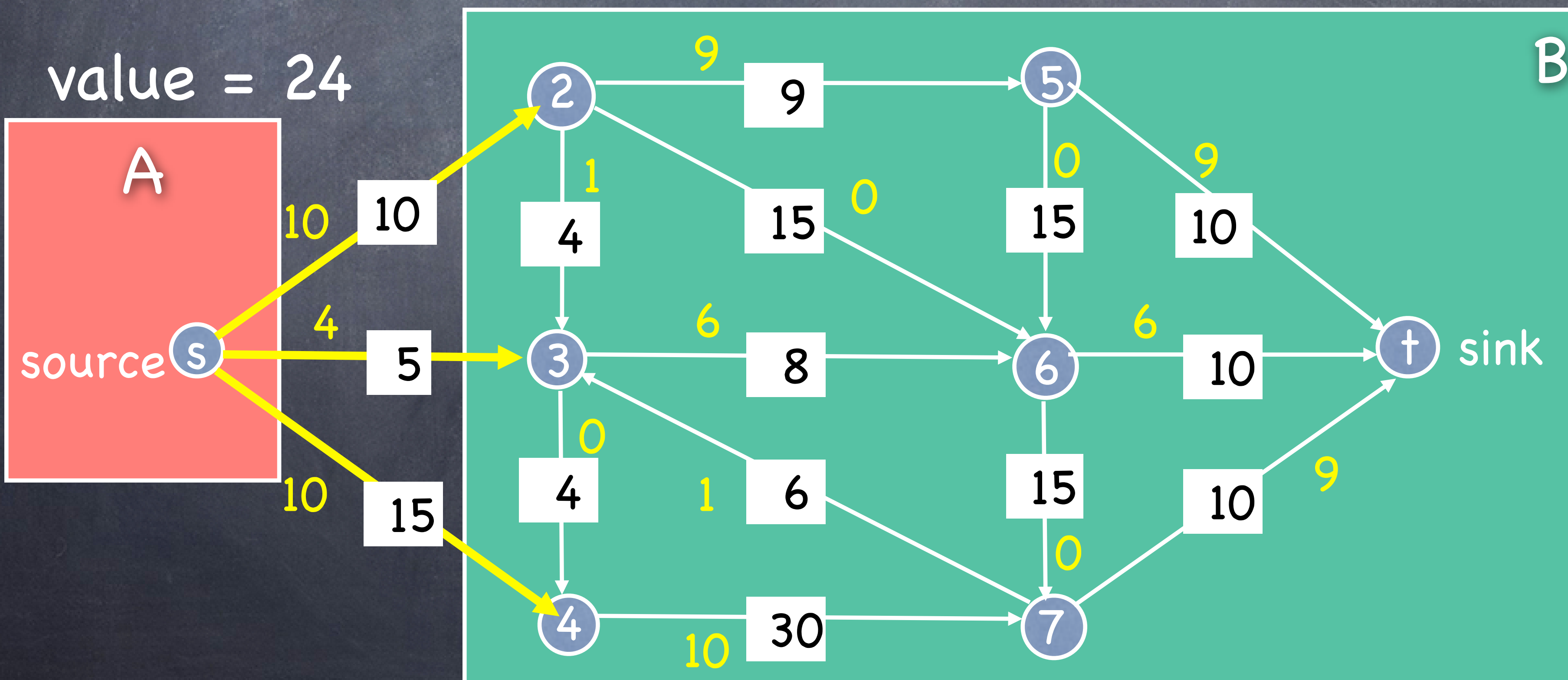


Flows and Cuts

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.

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

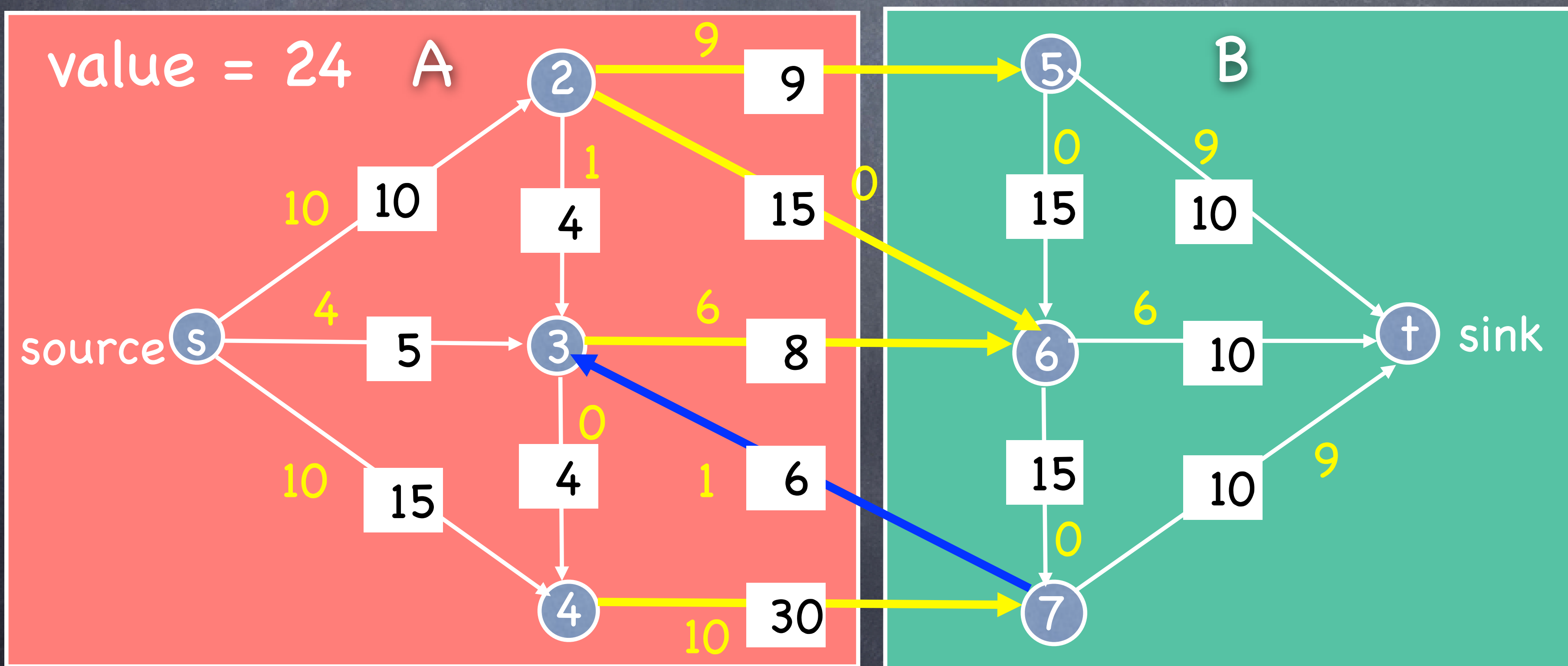
value = 24



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.

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

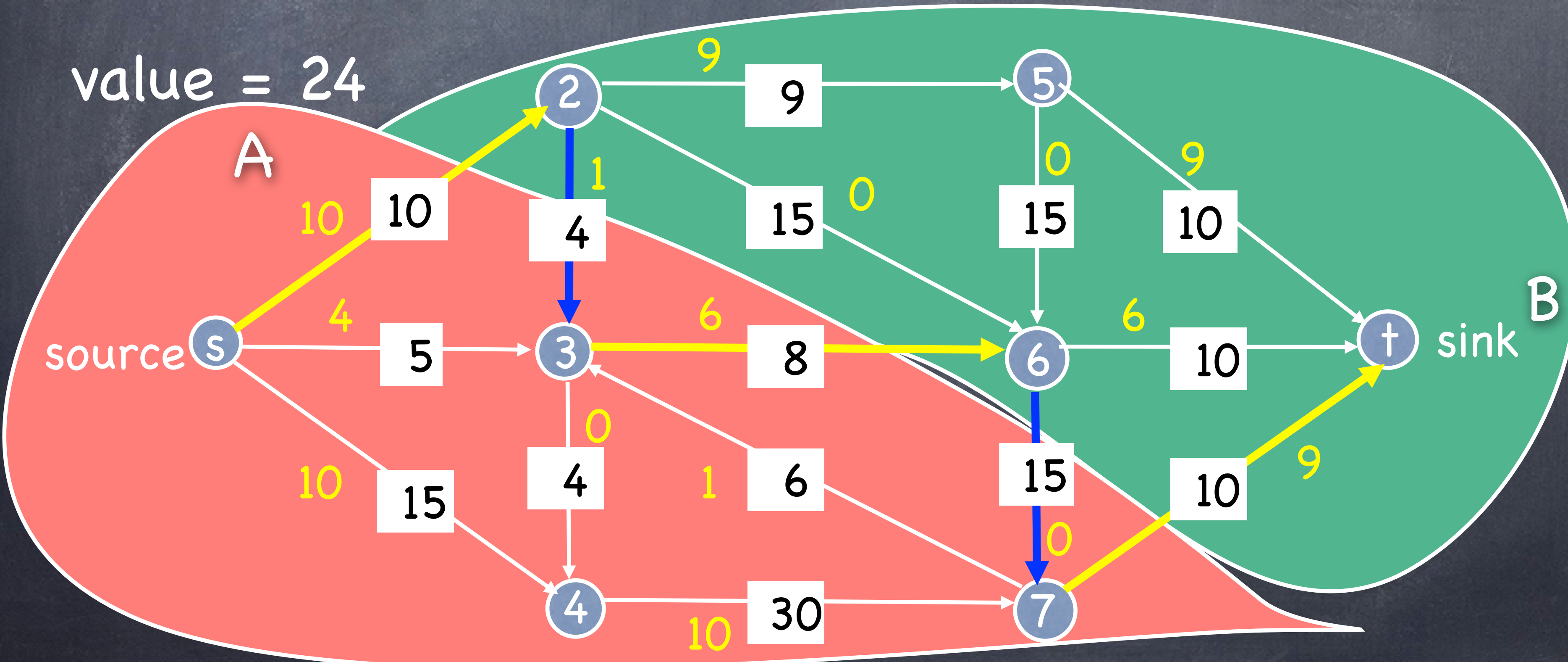


Flows and Cuts

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.

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

value = 24



Flows and Cuts

Another interpretation: we can measure the value of a flow by selecting any s-t cut, and looking at the net flow crossing the cut.

Proof

Flow value lemma. Let f be any flow, and let (A, B) be any s-t cut. Then $\sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e) = v(f)$.

Proof on board.

Important Corollary!

Flow value lemma. Let f be any flow, and let (A, B) be any s-t cut. Then $\sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e) = v(f)$.

Corollary. Let f be any flow, and let (A, B) be any s-t cut. Then $v(f) \le c(A, B)$.

(See examples on previous slides)

Proof as exercise

Important Corollary!

Corollary. Let f be any flow, and let (A, B) be any s-t cut. Then $v(f) \leq c(A, B)$.

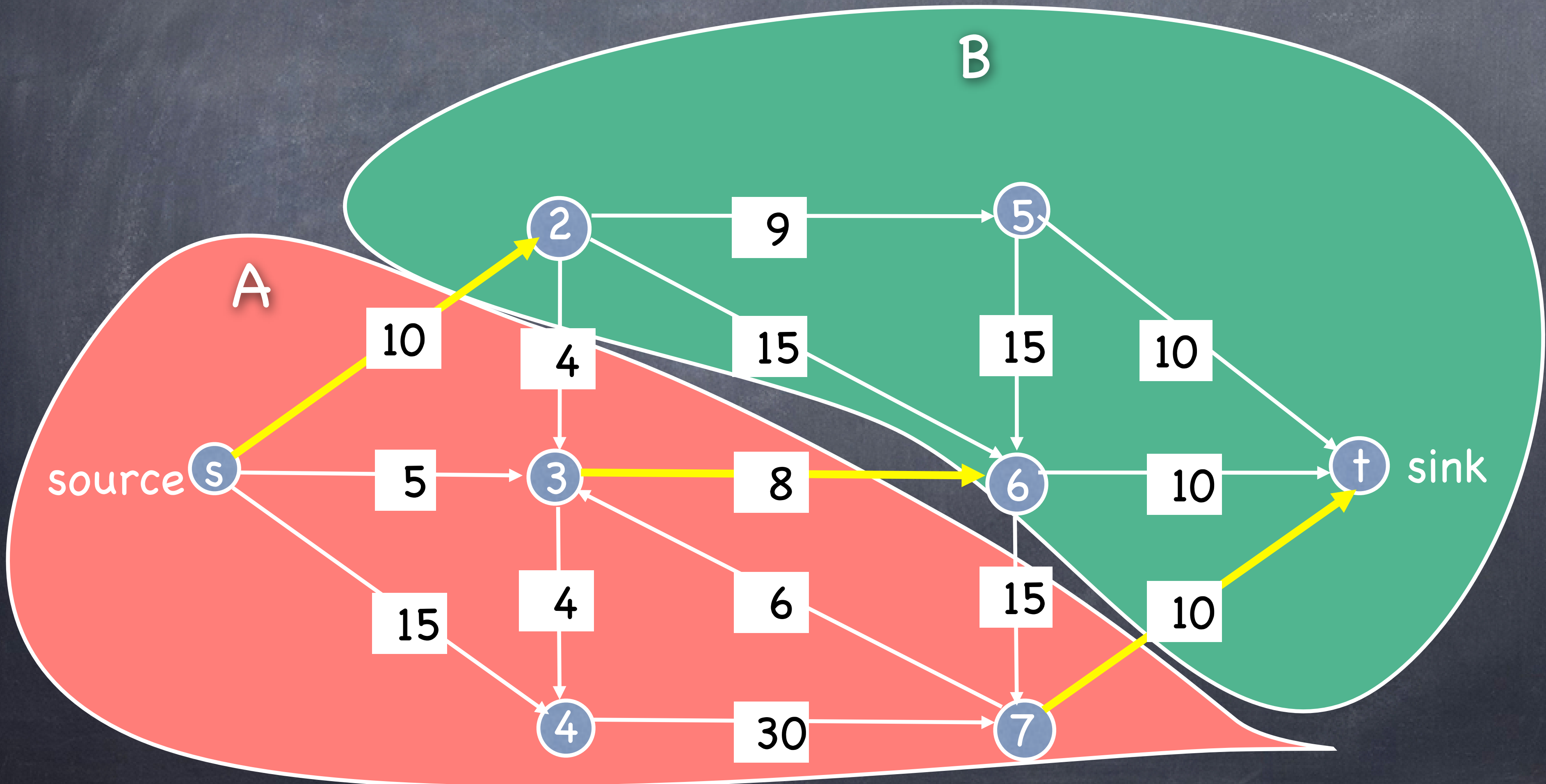
Interpretation: every cut gives an upper bound on the value of every flow, and hence the value of the maximum flow

What is the best (i.e. smallest) upper bound?

Minimum Cut Problem

Find an s-t cut of minimum capacity.

capacity = 10 + 8 + 10 = 28



Max-Flow Min-Cut Theorem

Theorem: Let f be a flow such that there are no s-t paths in the residual graph G_f. Let (A, B) be the s-t cut where A contains the nodes reachable from s in G_f, and B = V-A. Then v(f) = C(A, B), and f is a maximum flow and (A,B) is a minimum cut.

Corollary: Ford-Fulkerson returns a maximum flow.

Corollary: In any flow network, the value of the max flow is equal to the capacity of the min cut.

Proof

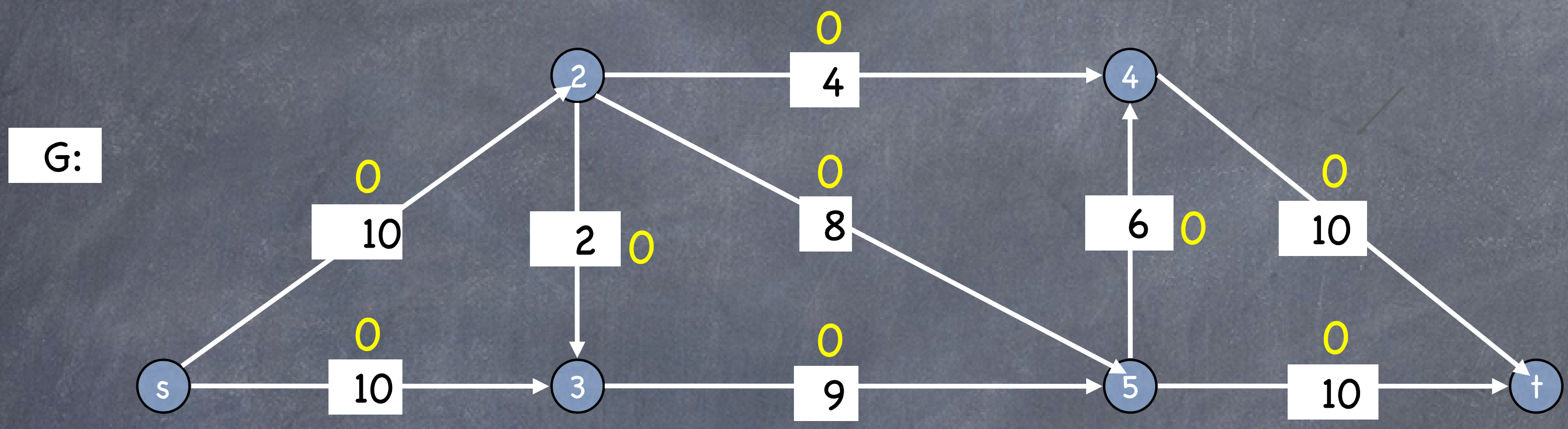
- Proof on board

Min-Cut

Another corollary: given a maximum flow, we can find a minimum cut in $O(m+n)$ time.

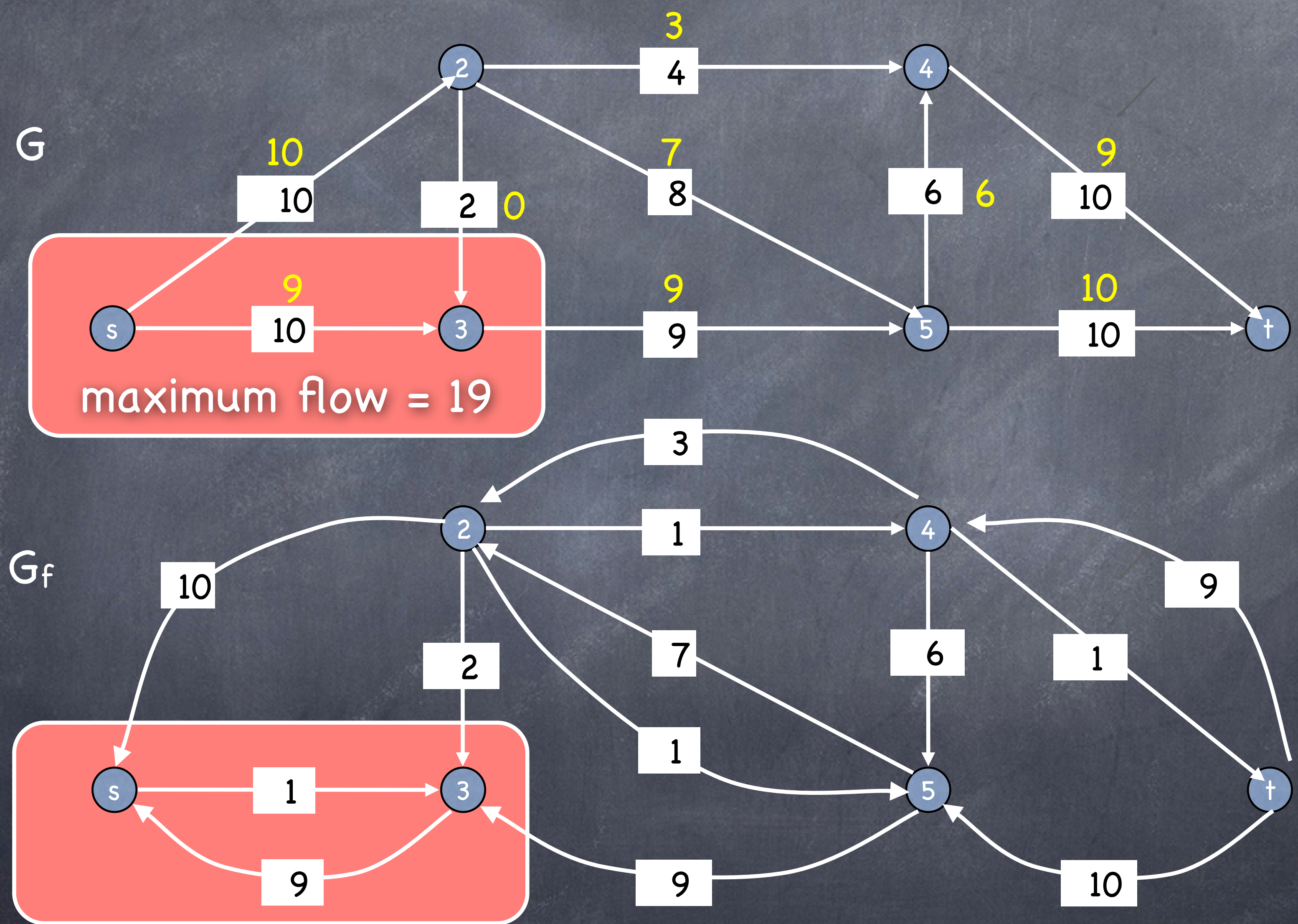
How?

Finding Min Cut



Flow value = 0

Finding Min Cut



Ford-Fulkerson Wrap-Up

We've now shown that F-F:
(1) runs in $O((m+n)OPT)$ time → pseudo-polynomial
(2) finds an integer-valued flow (if capacities are integers)
(3) finds a maximum flow

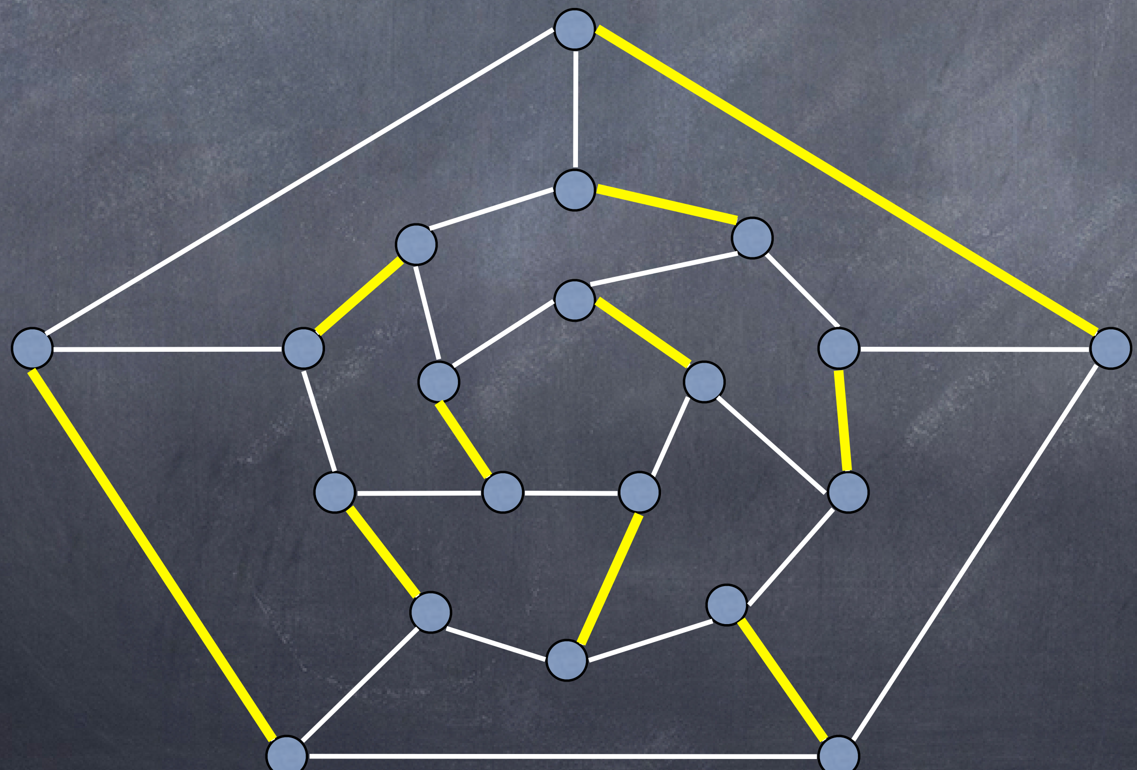
By choosing good augmenting paths, F-F can be improved to run $O(m^2 \log C)$ time, where C is the capacity of any cut, and hence an upper bound on OPT → polynomial

Other max-flow algorithms run in $O(n^2m)$ or $O(n^3)$ time → strongly polynomial

OK! But what are they good for???

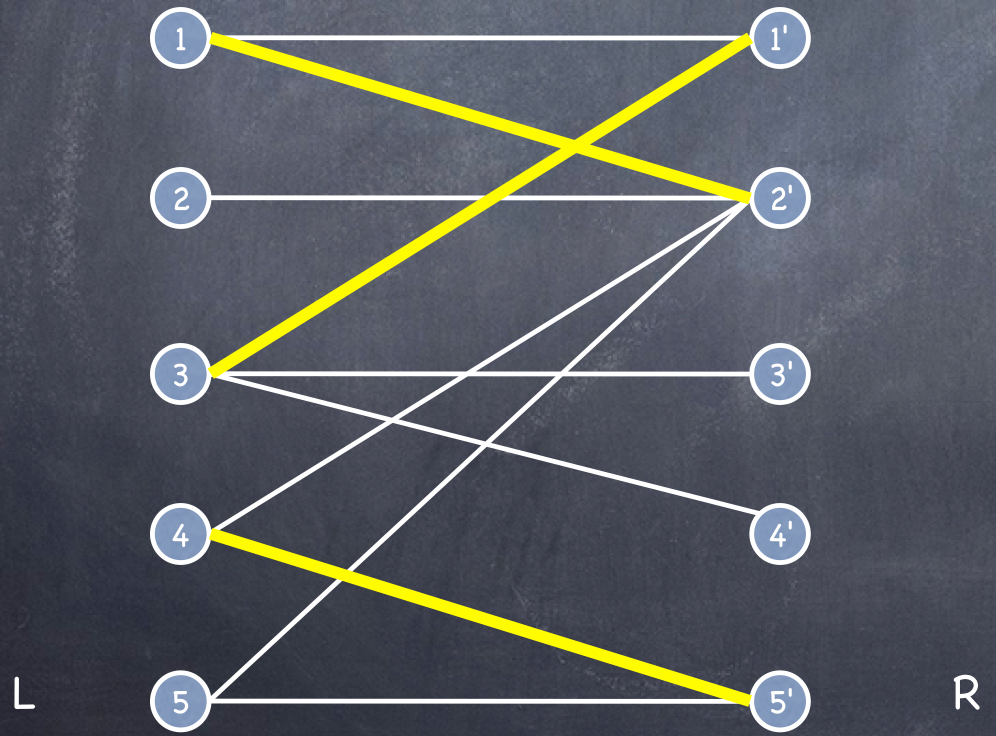
Matching

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



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.



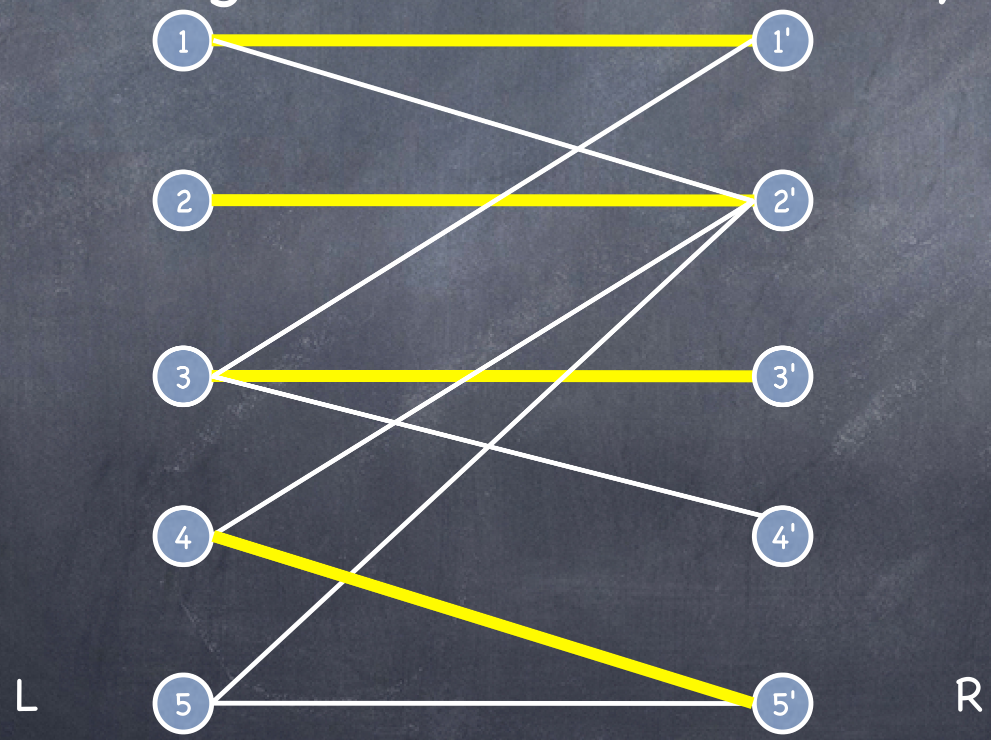
matching

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

Is this the max 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.

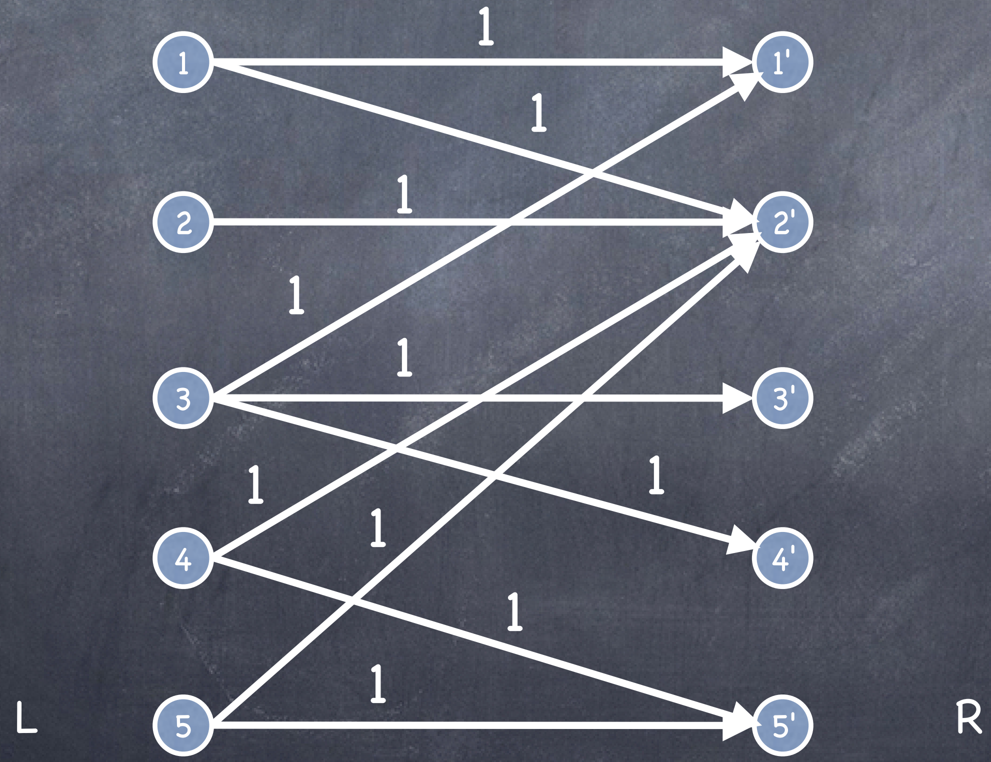


max matching

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

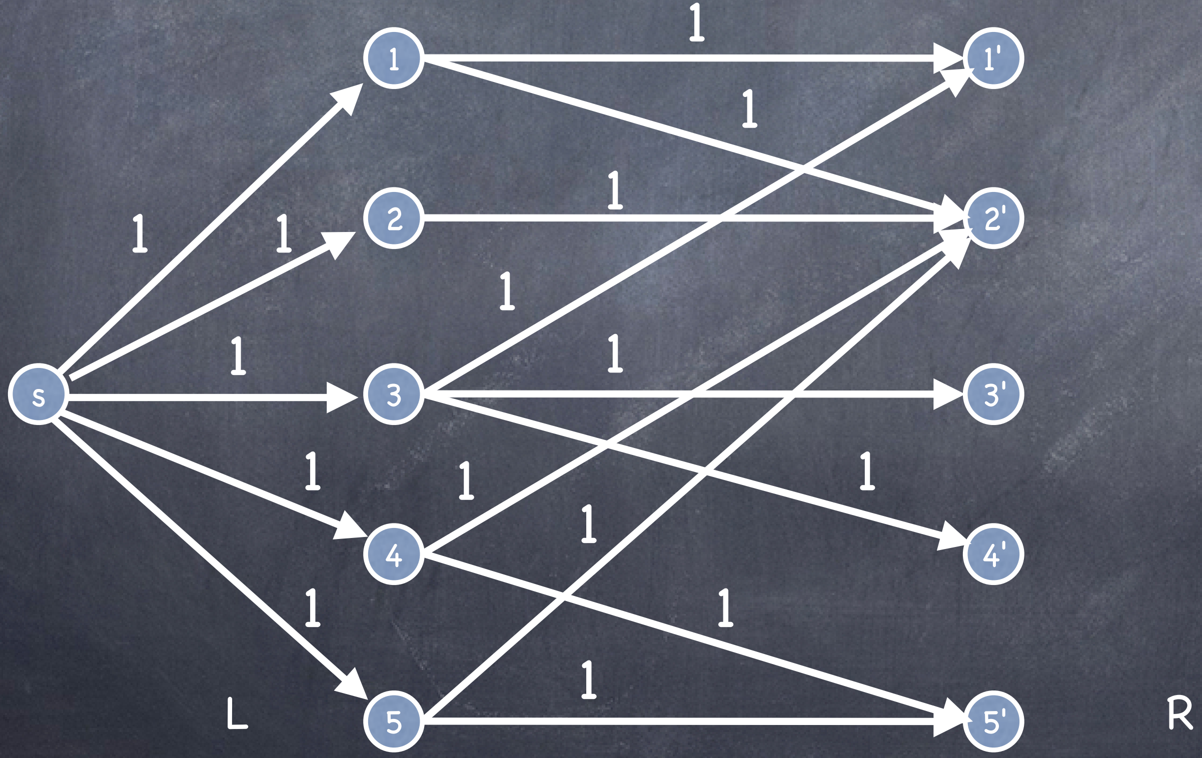
Bipartite Matching

- Max flow formulation.
  - Direct all edges from L to R and assign capacity of 1.



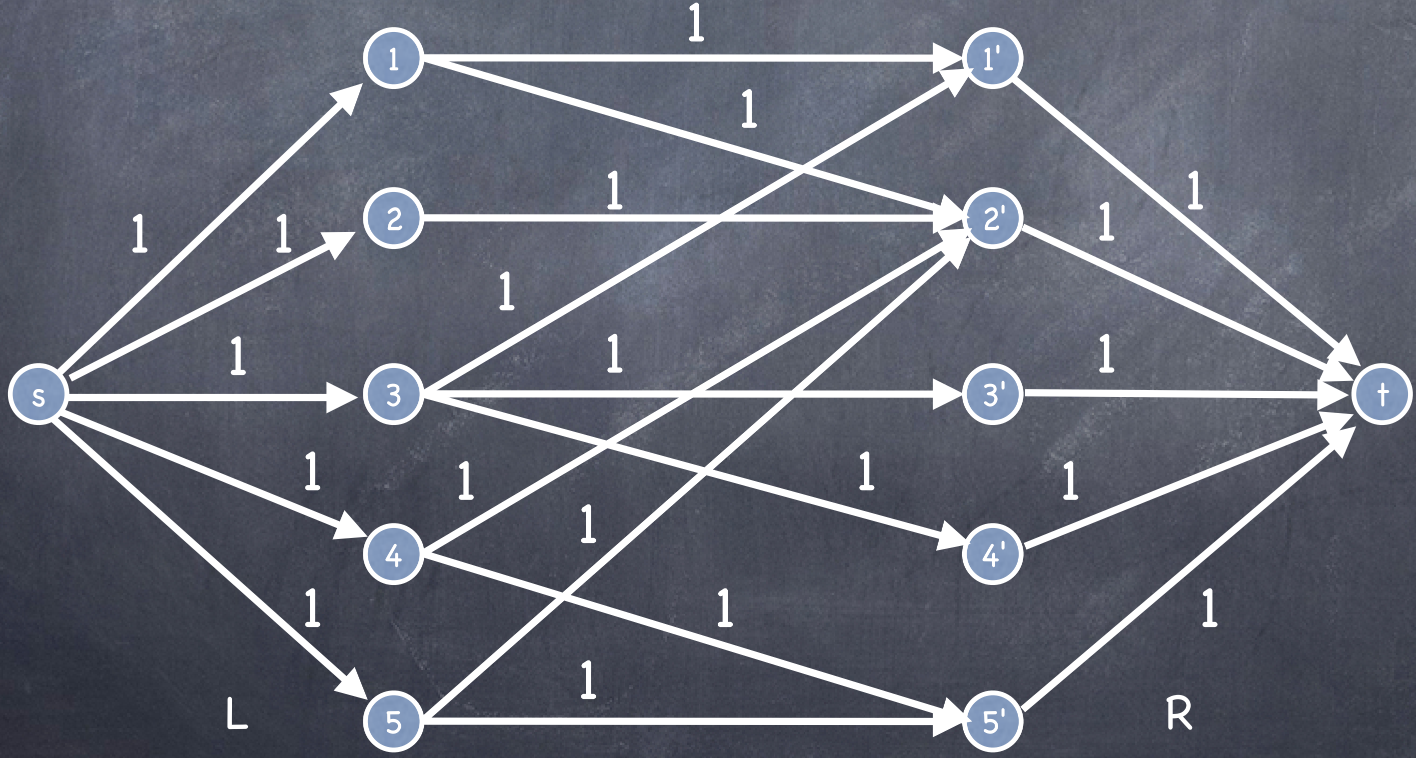
Bipartite Matching

- Max flow formulation.
  - Add source s, and unit capacity edges from s to each node in L.



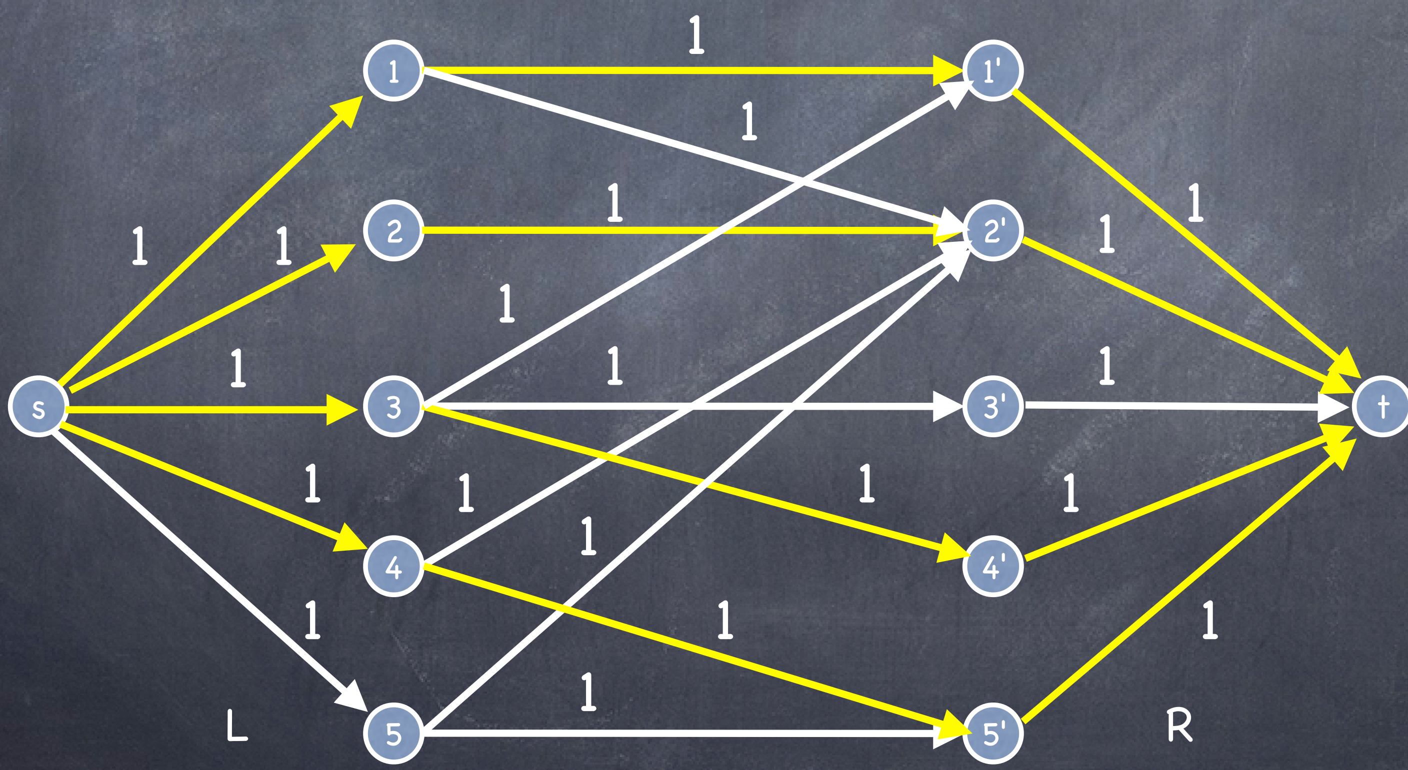
Bipartite Matching

- Max flow formulation.
  - Add sink t, and unit capacity edges from each node in R to t.



Bipartite Matching

- Max flow formulation.
  - Solve max flow problem.
  - Claim: edges between L and R with flow = 1 identify max matching.



Proof

Show there is a bijection between a matching M in the original graph, and a flow f in the new graph, and that $v(f) = |M|$.

Thus, a maximum flow is a maximum matching.

Details: exercise

Next Time

- More flow applications!