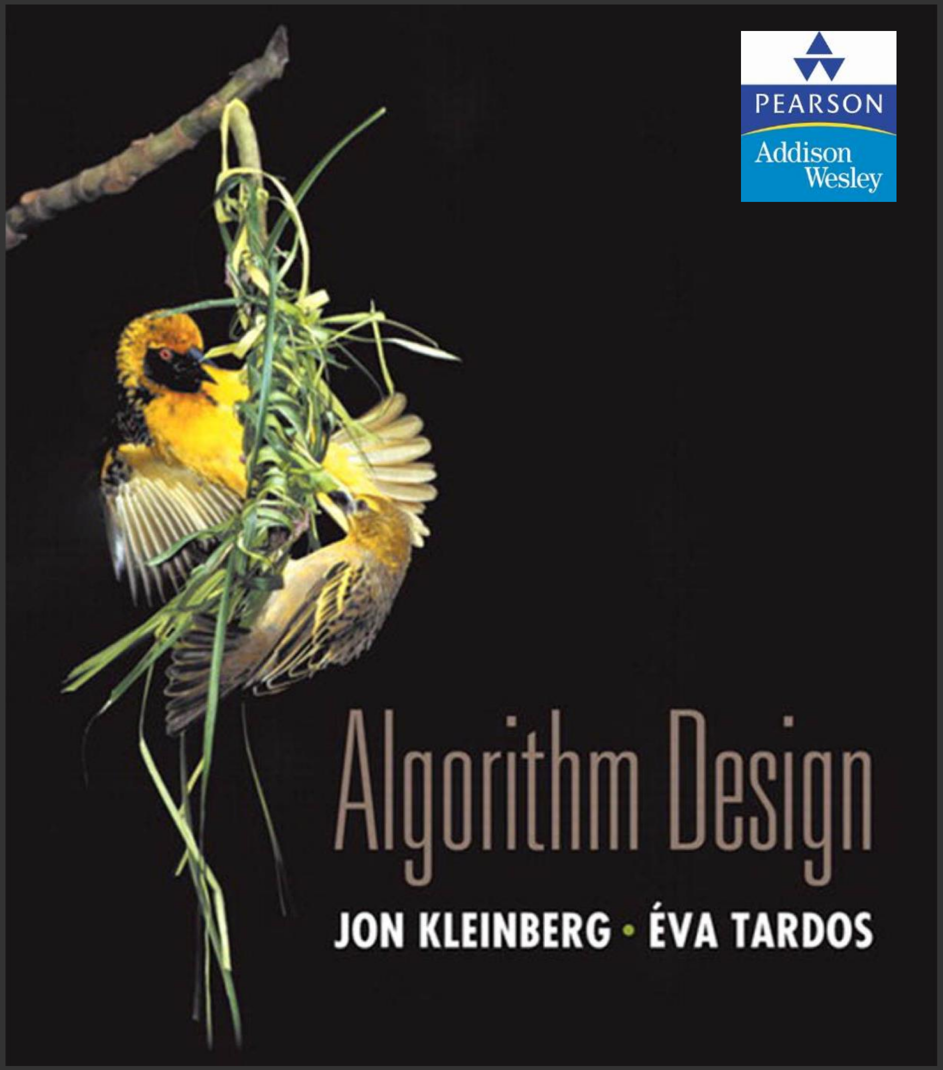


7. Network Flow II

- *bipartite matching*
- *disjoint paths*
- *image segmentation*
- *baseball elimination*

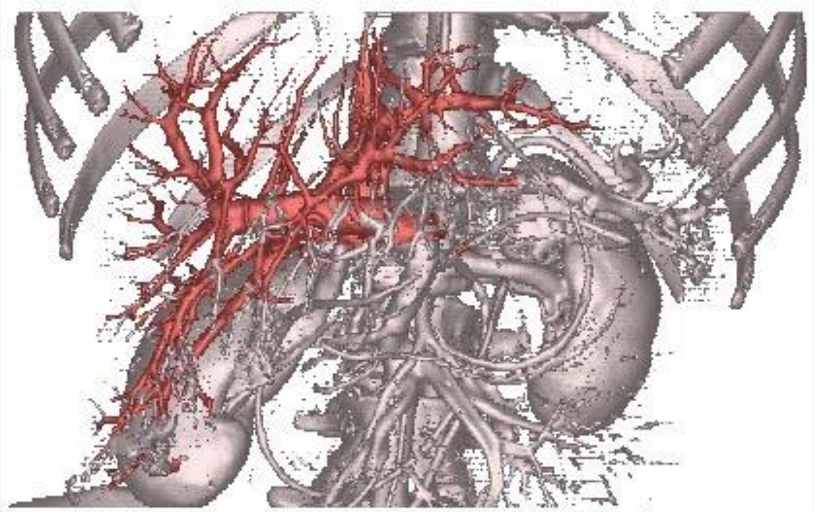
Lecture slides by Kevin Wayne

http://www.cs.princeton.edu/~wayne/kleinberg-tardos

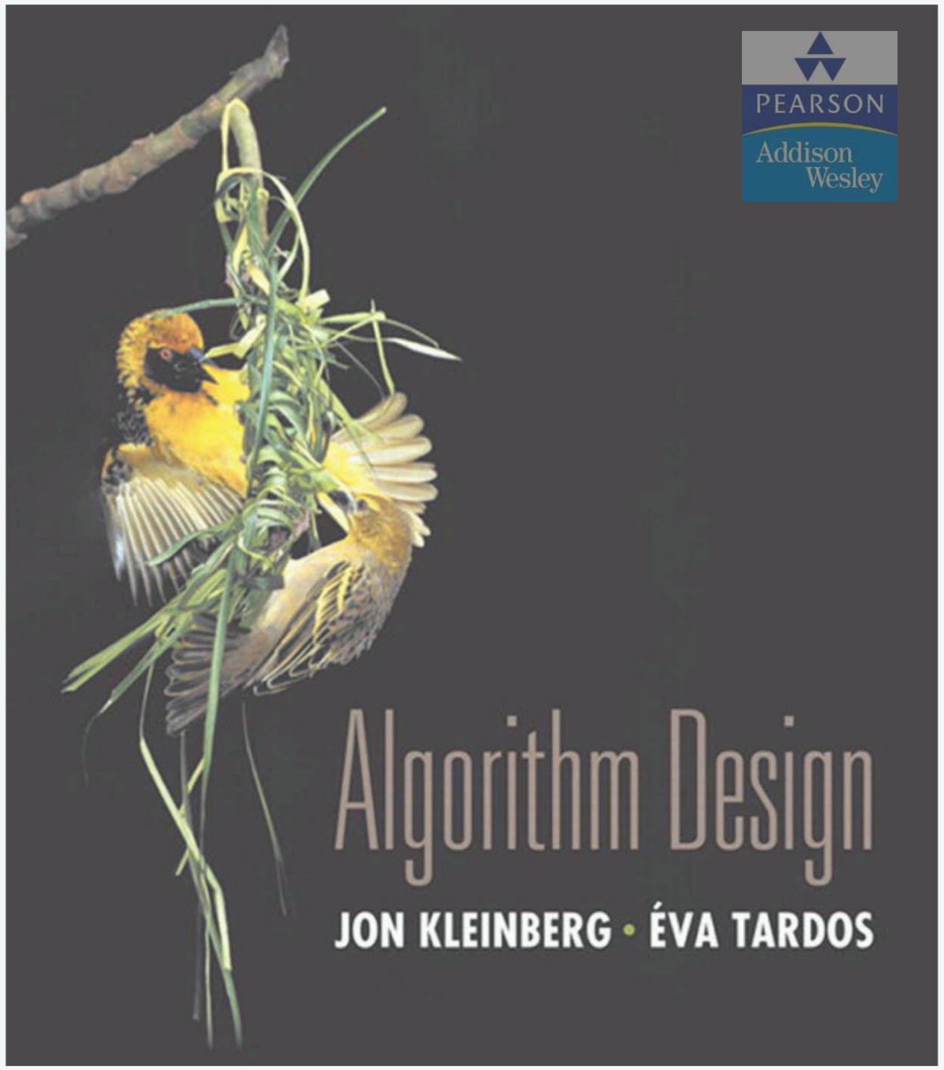
Max-flow and min-cut applications

Max-flow and min-cut problems are widely applicable model.

- Data mining.
- Open-pit mining.
- Bipartite matching.
- Network reliability.
- Baseball elimination.
- Image segmentation.
- Network connectivity.
- Markov random fields.
- Distributed computing.
- Security of statistical data.
- Egalitarian stable matching.
- Network intrusion detection.
- Multi-camera scene reconstruction.
- Sensor placement for homeland security.
- Many, many, more.



liver and hepatic vascularization segmentation



SECTION 7.5

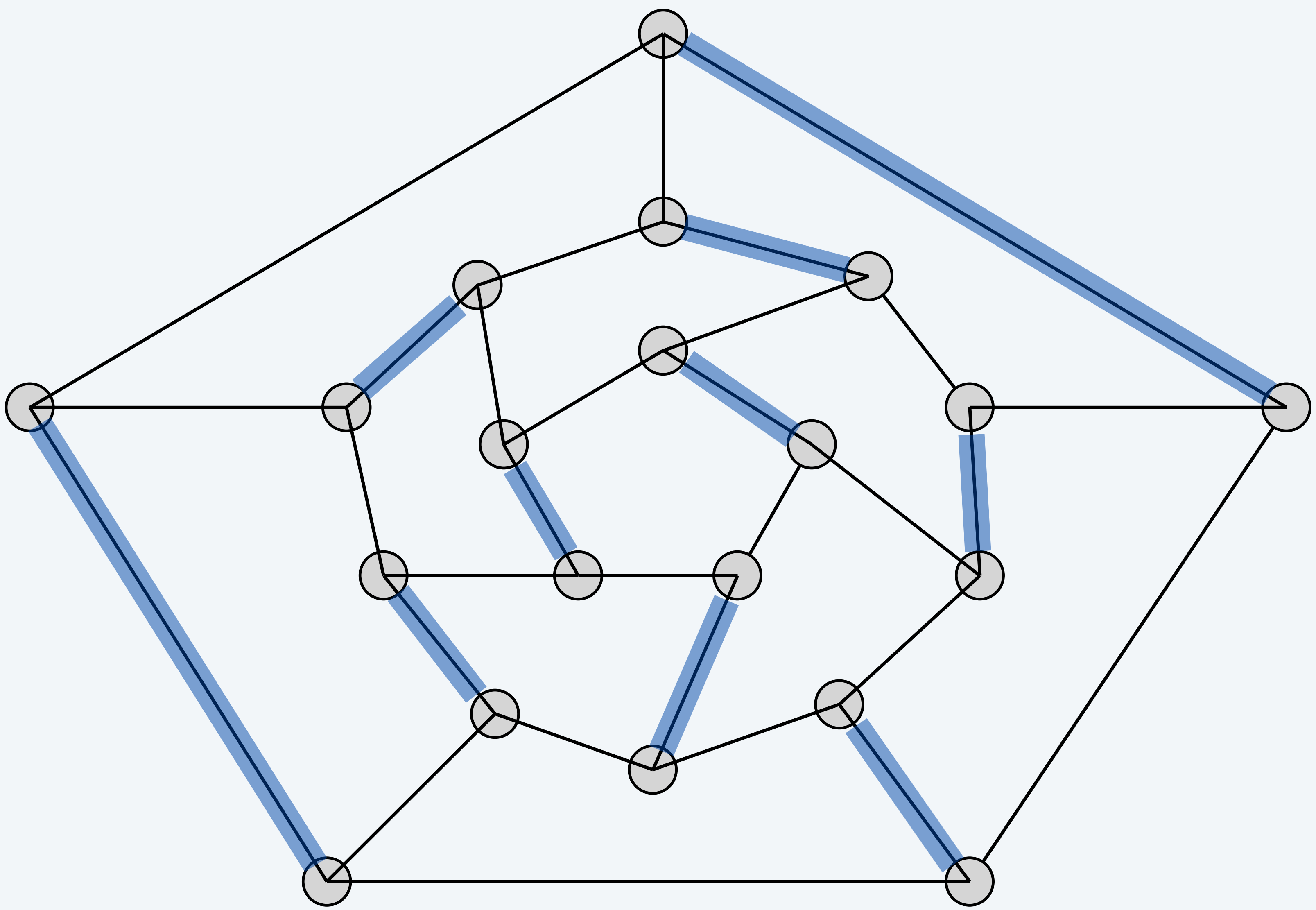
7. NETWORK FLOW II

- *bipartite matching*
- *disjoint paths*
- *image segmentation*
- *baseball elimination*

Matching

Def. Given an undirected graph $G = (V, E)$, subset of edges $M \subseteq E$ is a matching if each node appears in at most one edge in M.

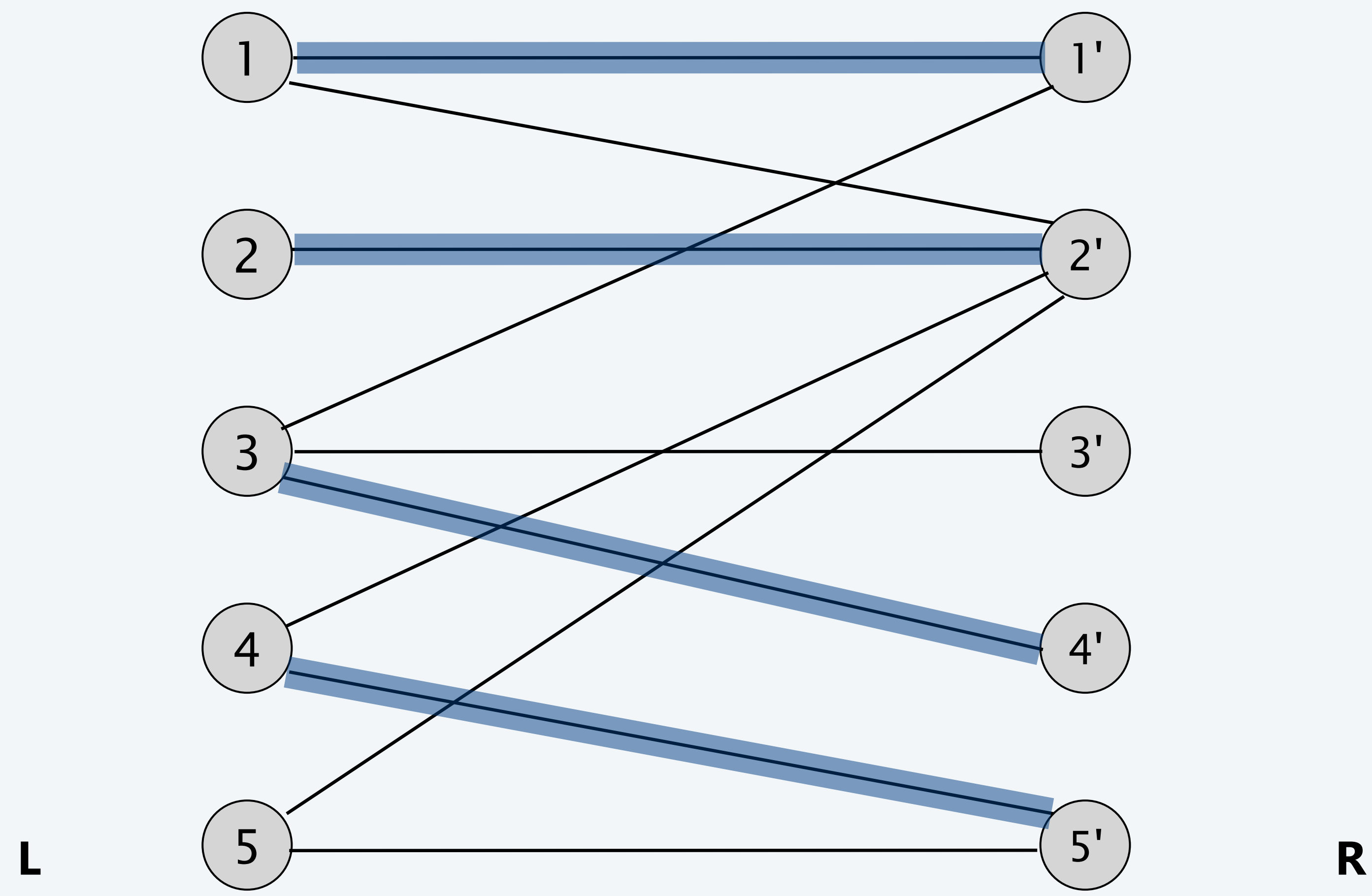
Max matching. Given a graph G, find a max-cardinality matching.



Bipartite matching

Def. A graph G is **bipartite** if the nodes can be partitioned into two subsets L and R such that every edge connects a node in L with a node in R.

Bipartite matching. Given a bipartite graph $G = (L \cup R, E)$, find a max-cardinality matching.

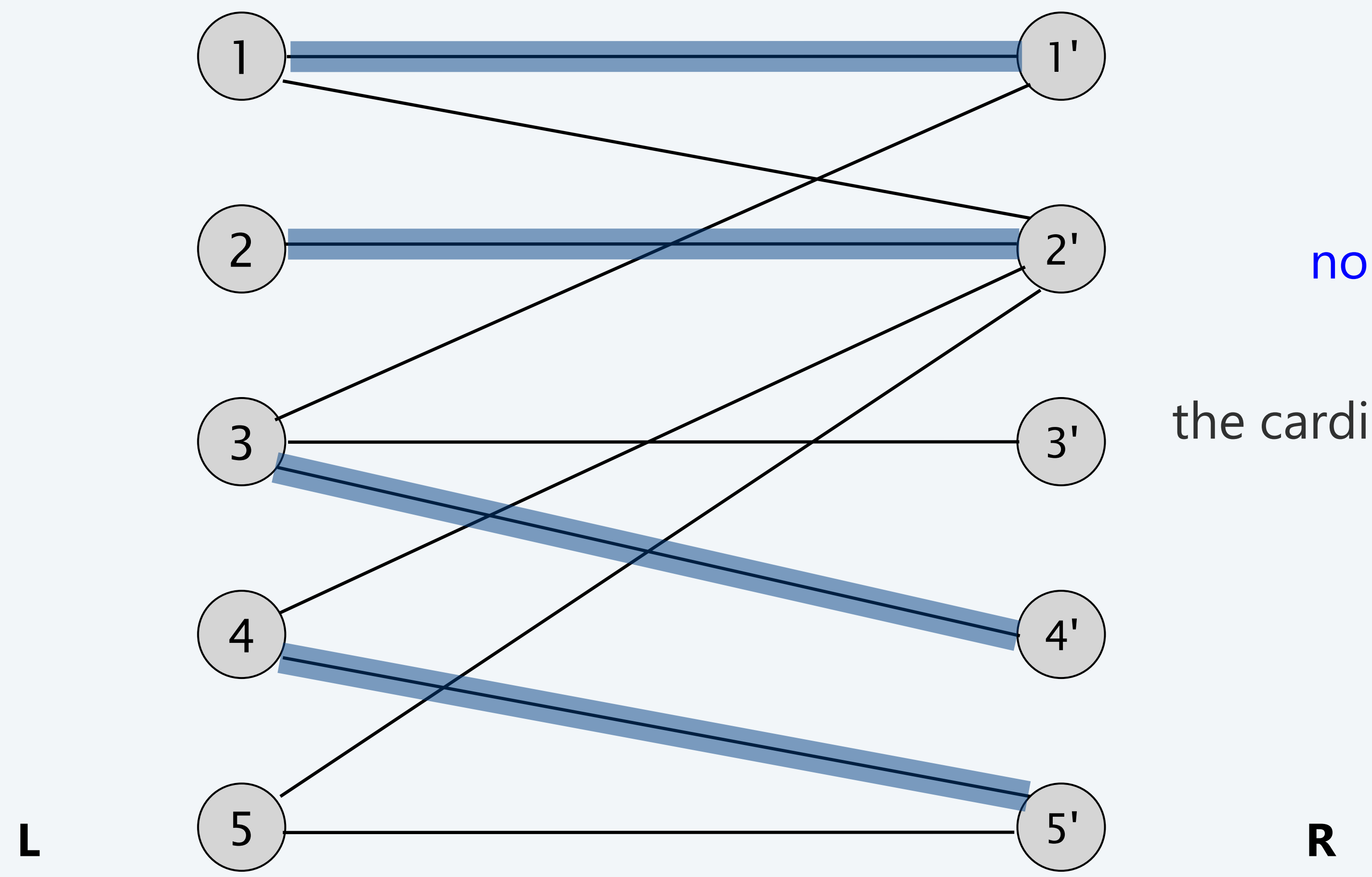


matching: 1-1', 2-2', 3-4', 4-5'

Perfect matching (in bipartite graphs)

Def. Given a graph $G = (V, E)$, a subset of edges $M \subseteq E$ is a perfect matching if each node appears in exactly one edge in M.

Perfect matching problem. Given a bipartite graph $G = (L \cup R, E)$, find a perfect matching or correctly report it does not exist.



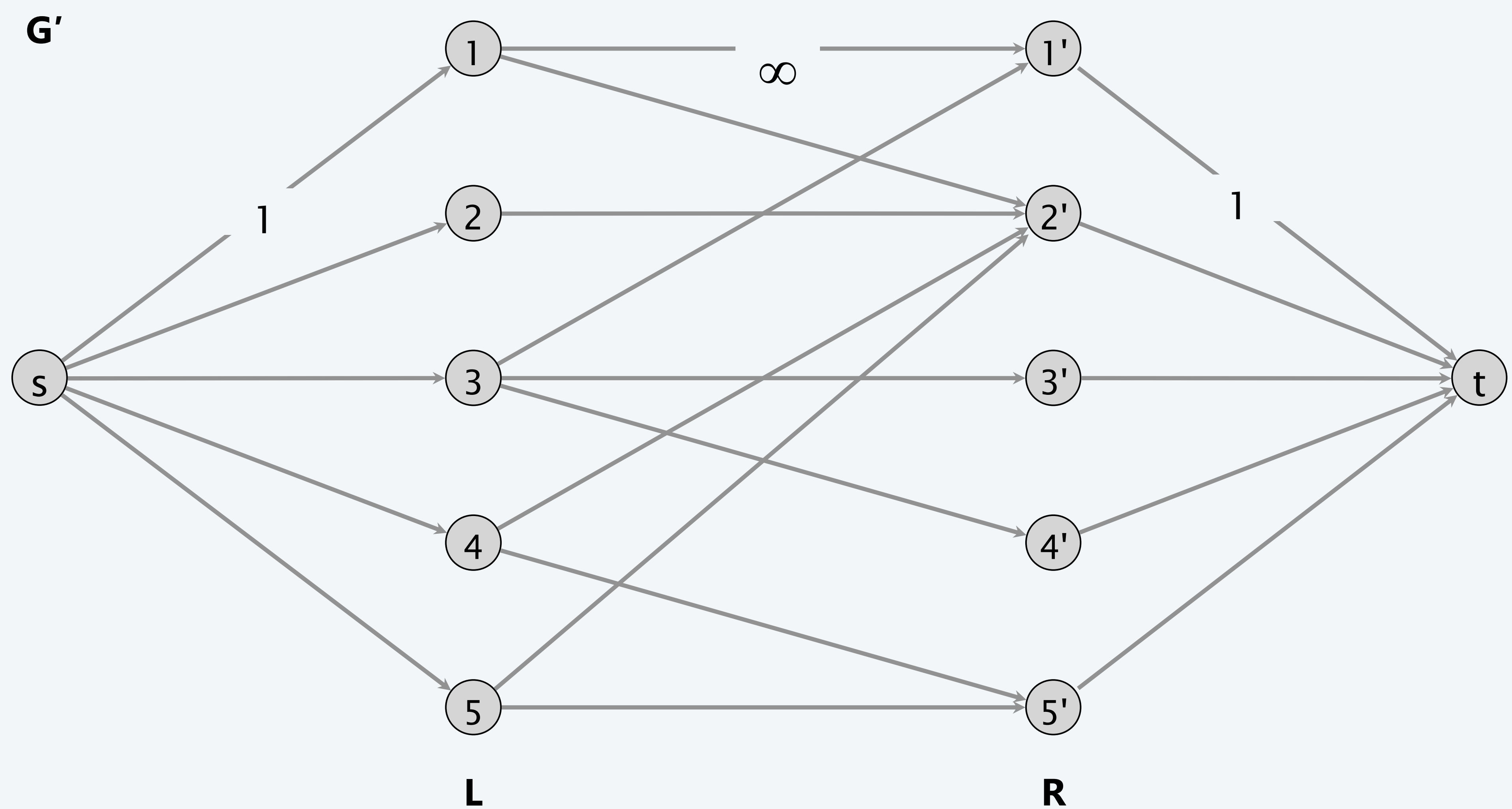
no perfect matching here:
perfect matching iff
the cardinality of maximum matching
is =|L|=|R|

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

Bipartite matching: max-flow formulation

Formulation.

- Create digraph $G' = (L \cup R \cup \{s, t\},\ E')$.
- Direct all edges from L to R, and assign infinite (or unit) capacity.
- Add unit-capacity edges from s to each node in L.
- Add unit-capacity edges from each node in R to t.



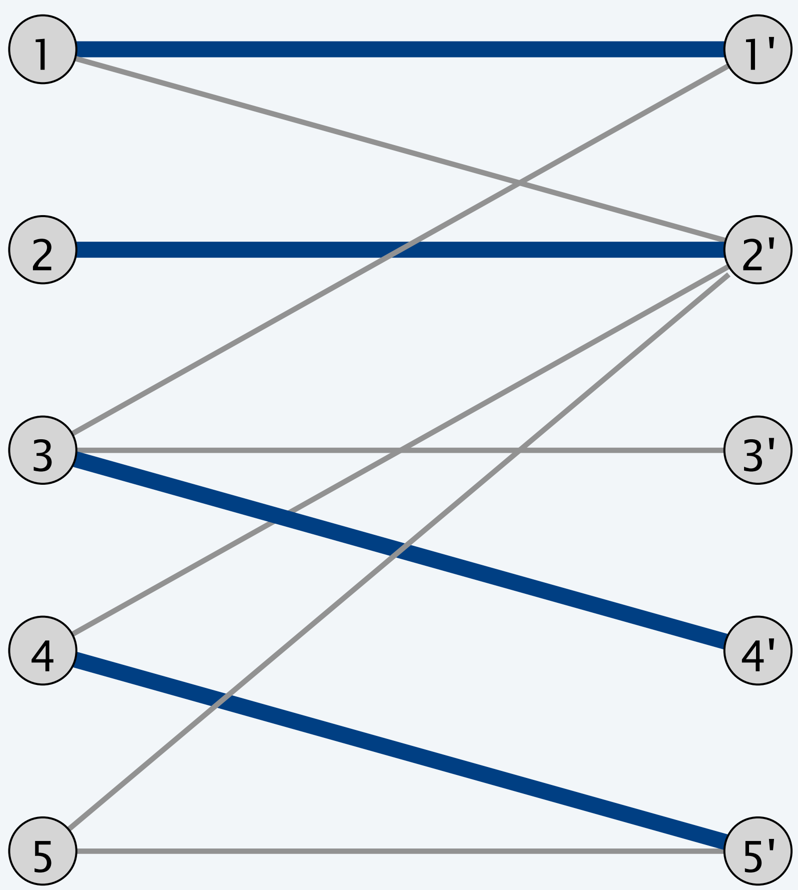
Max-flow formulation: proof of correctness

Theorem. 1–1 correspondence between matchings of cardinality k in G and integral flows of value k in G'.

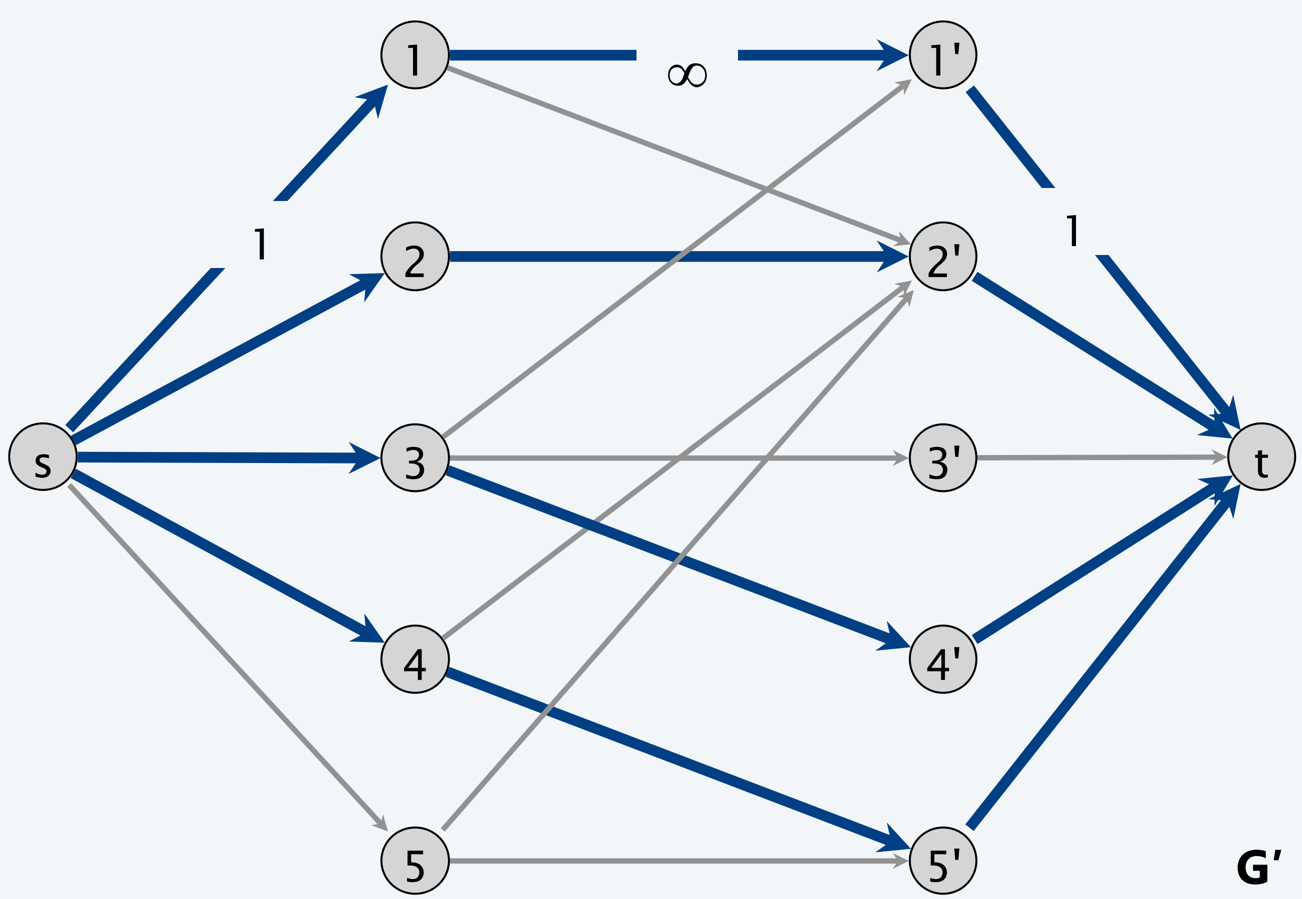
for each edge e: $f(e) \in \{ 0, 1 \}$

Pf. $\Rightarrow$

- Let M be a matching in G of cardinality k.
- Consider flow f that sends 1 unit on each of the k corresponding paths.
- f is a flow of value k. ▪



G



G′

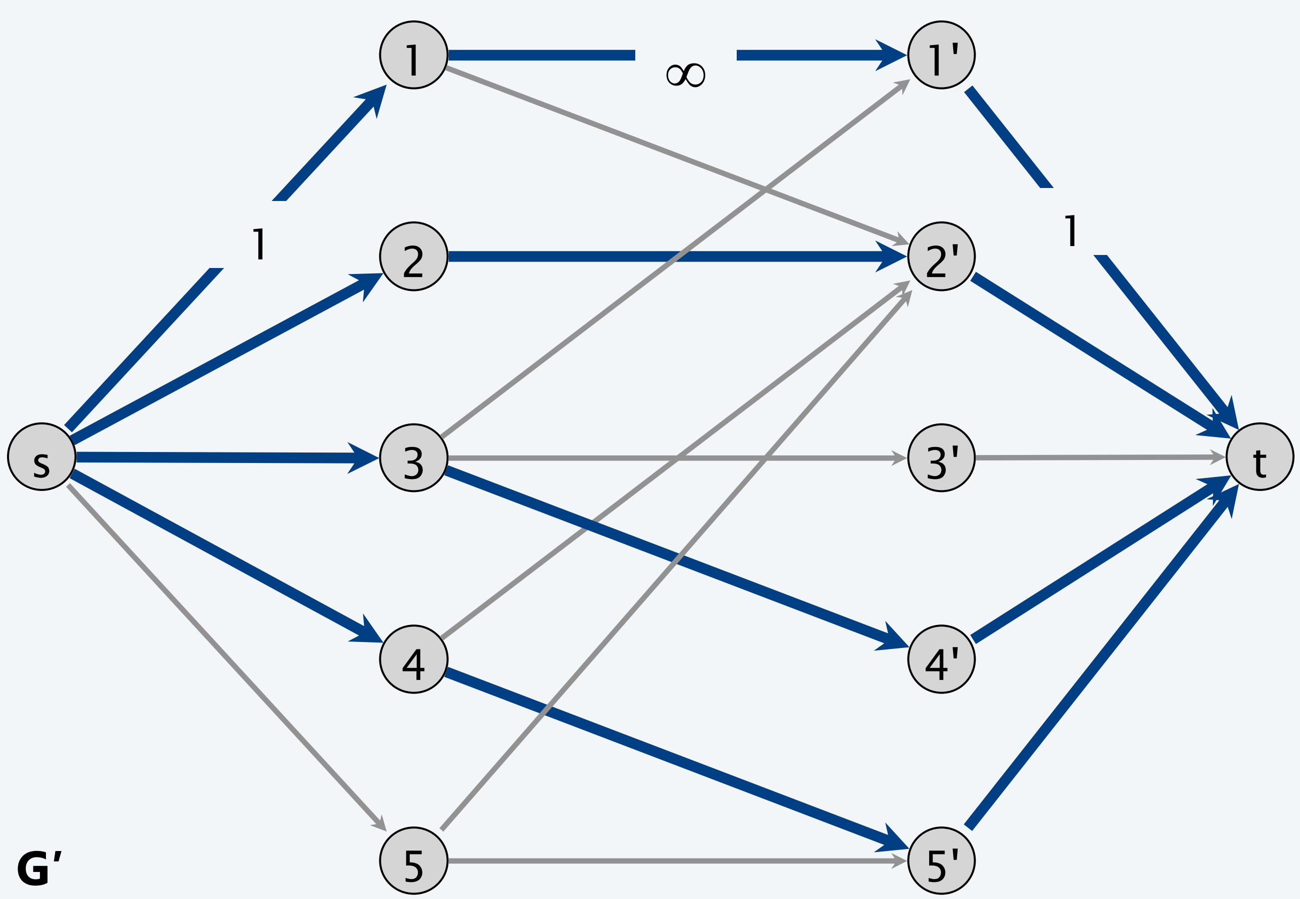
Max-flow formulation: proof of correctness

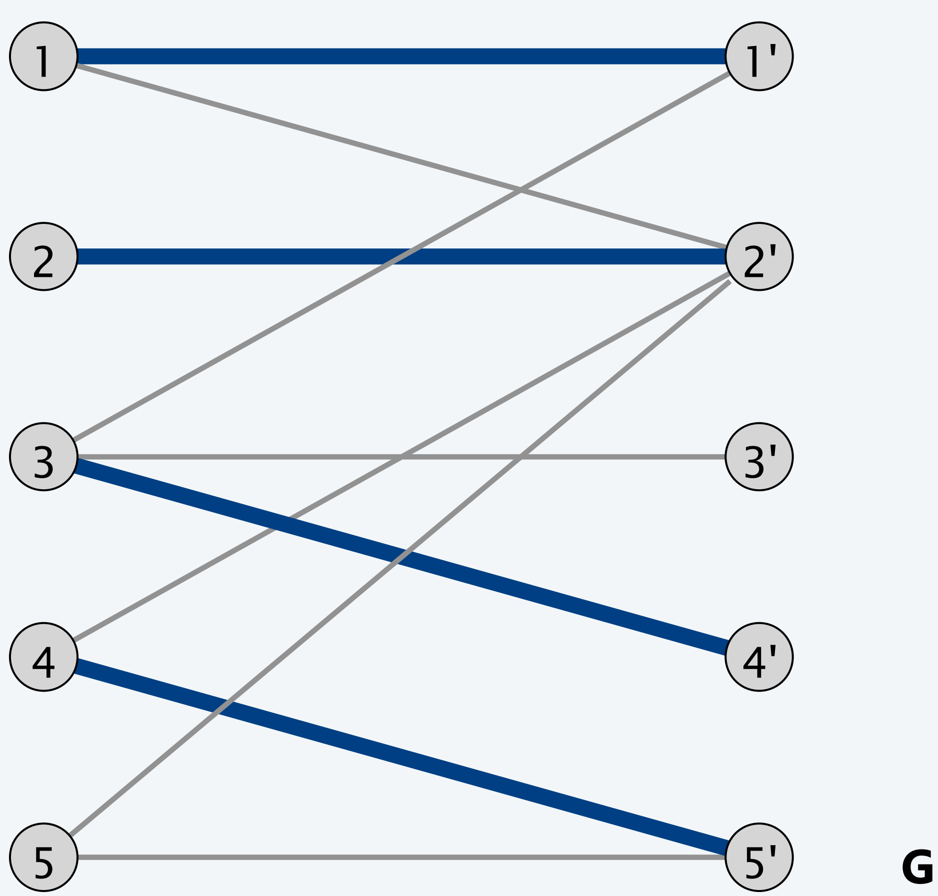
Theorem. 1–1 correspondence between matchings of cardinality k in G and integral flows of value k in G'.

for each edge e: $f(e) \in \{ 0, 1 \}$

Pf. $\Leftarrow$

- Let f be an integral flow in G' of value k.
- 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
  - $|M| = k$: apply flow-value lemma to cut $(L \cup \{s\}, R \cup \{t\})$ ▪





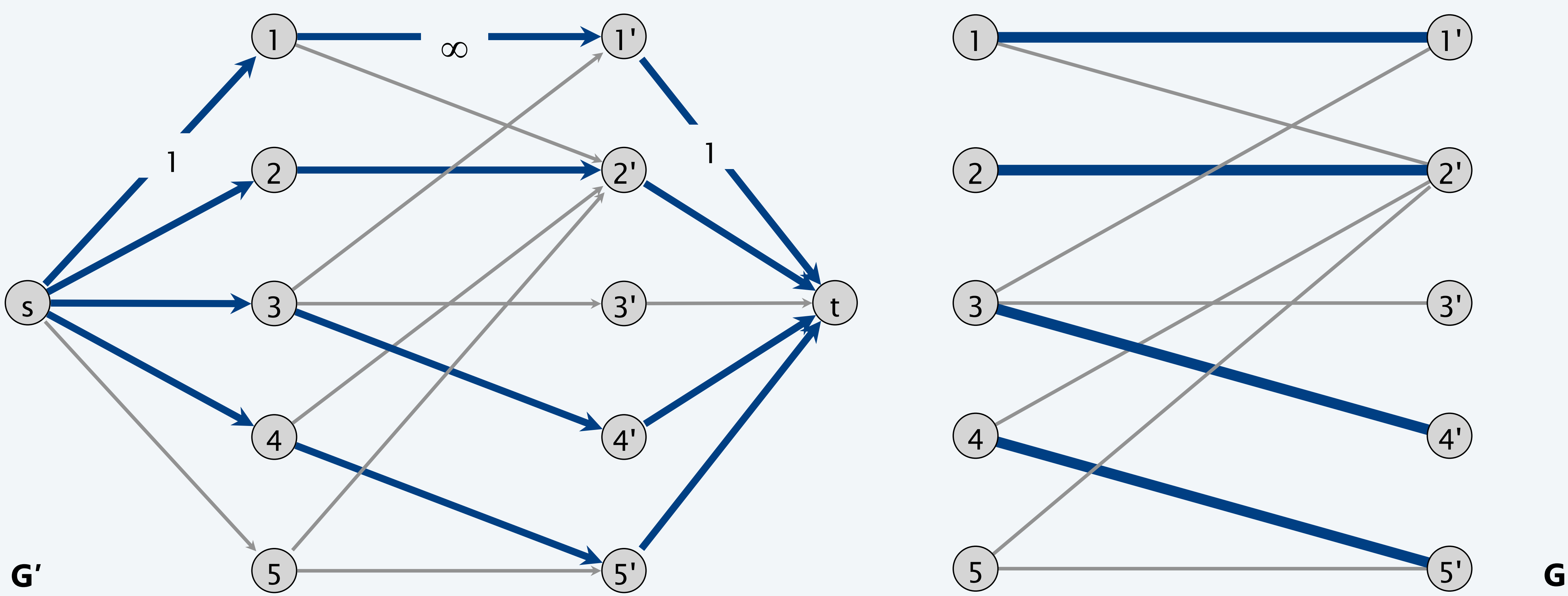
Max-flow formulation: proof of correctness

Theorem. 1–1 correspondence between matchings of cardinality k in G and integral flows of value k in G'.

Corollary. Can solve bipartite matching problem via max-flow formulation.

Pf.

- Integrality theorem $\Rightarrow$ there exists a max flow f^* in G' that is integral.
- 1–1 correspondence $\Rightarrow$ f^* corresponds to max-cardinality matching. ▪



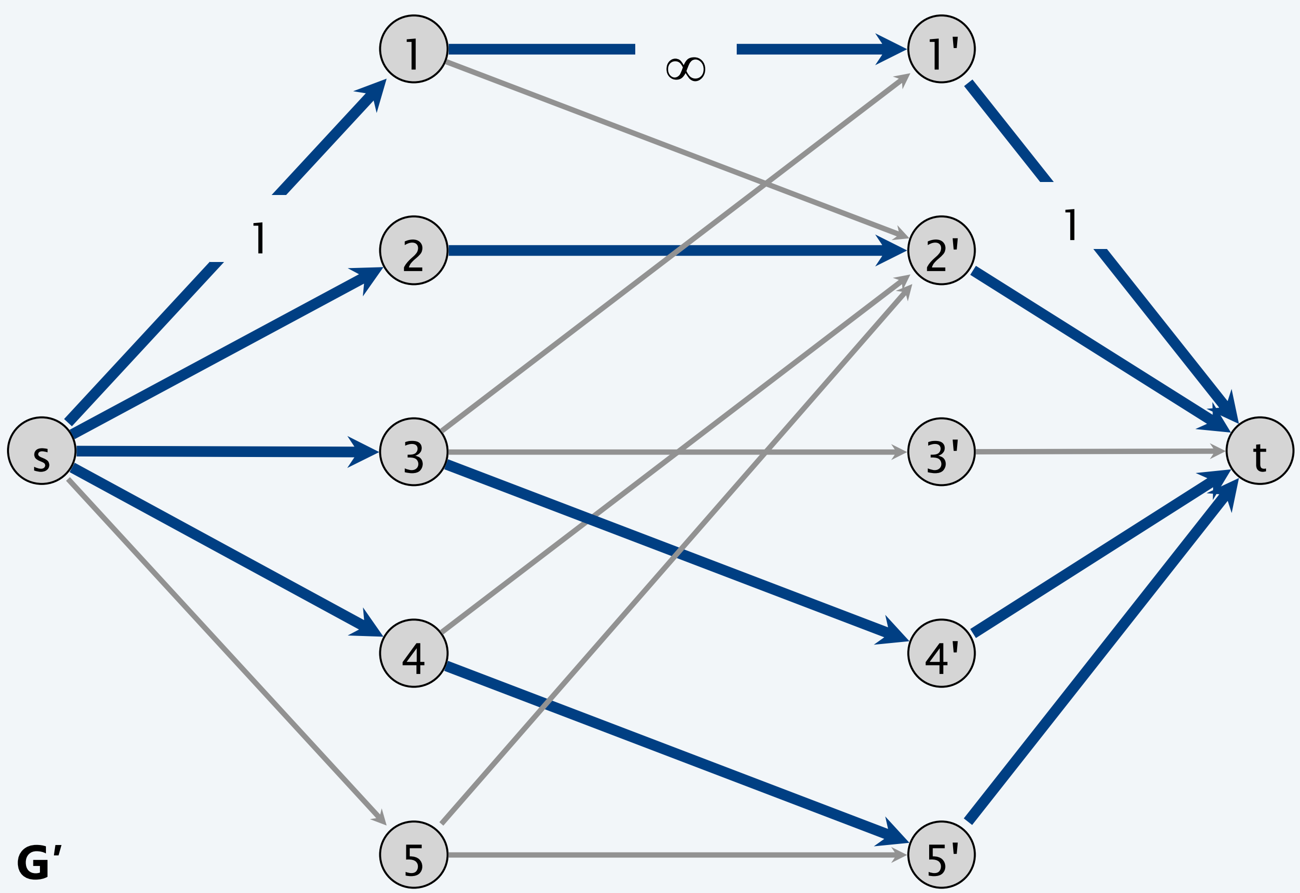
Max-flow formulation: running time

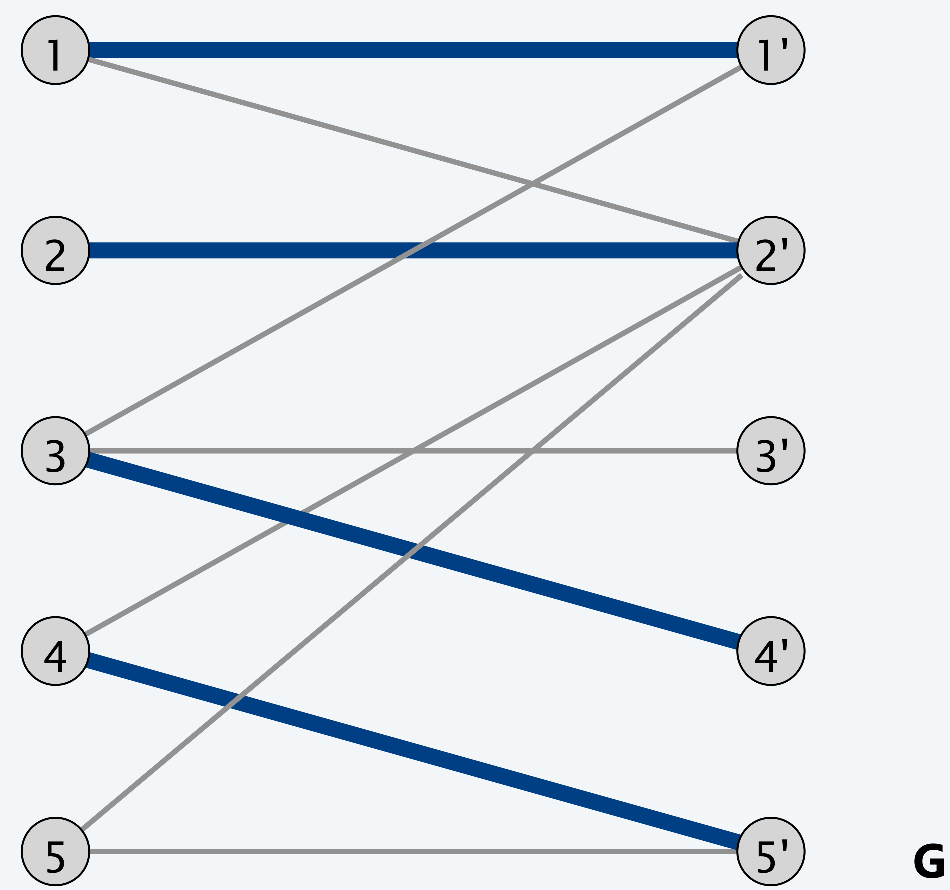
Theorem. 1–1 correspondence between matchings of cardinality k in G and integral flows of value k in G'.

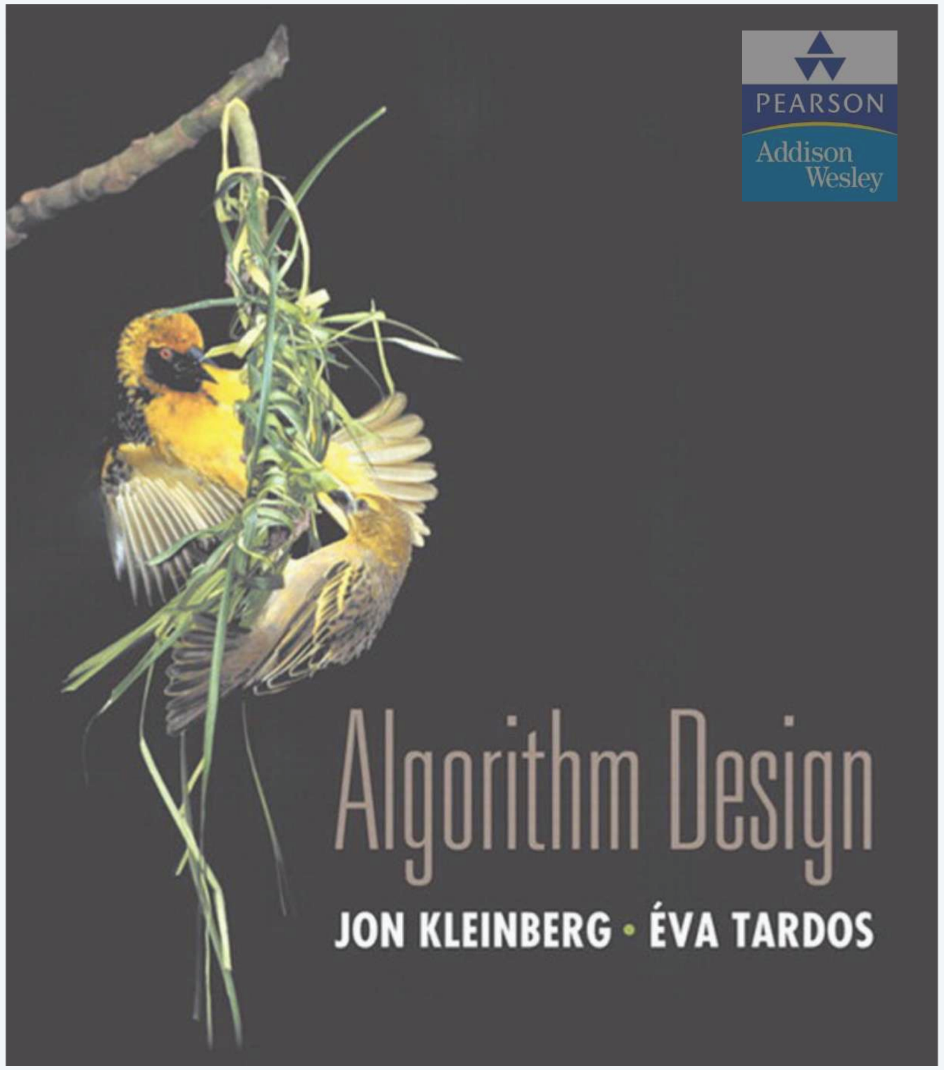
Corollary. Can solve bipartite matching problem via max-flow formulation.

Running time:

- Using Ford-Fulkerson:
- $\leq n$ augmentations $\Rightarrow O(mn)$ time.







SECTION 7.6

7. NETWORK FLOW II

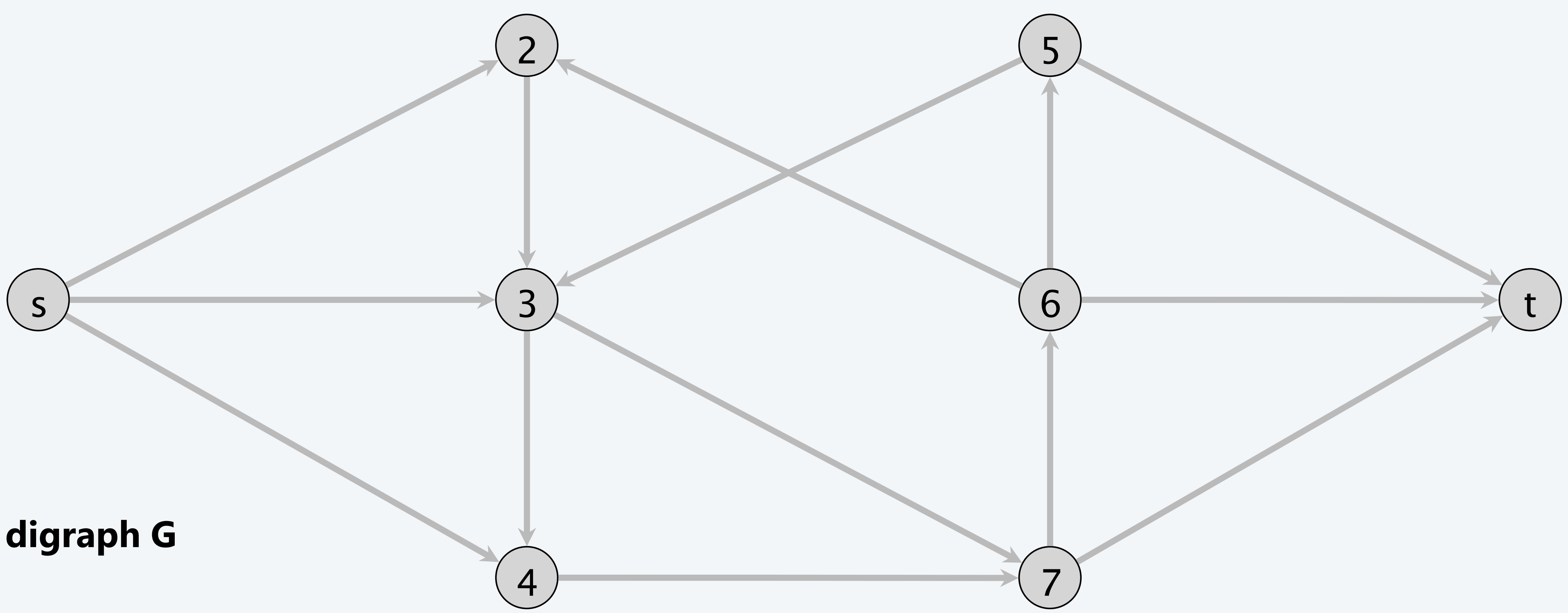
- *bipartite matching*
- *disjoint paths*
- *image segmentation*
- *baseball elimination*

Edge-disjoint paths

Def. Two paths are edge-disjoint if they have no edge in common.

Edge-disjoint paths problem. Given a digraph $G = (V, E)$ and two nodes s and t, find the max number of edge-disjoint $s \leadsto t$ paths.

Ex. Communication networks.



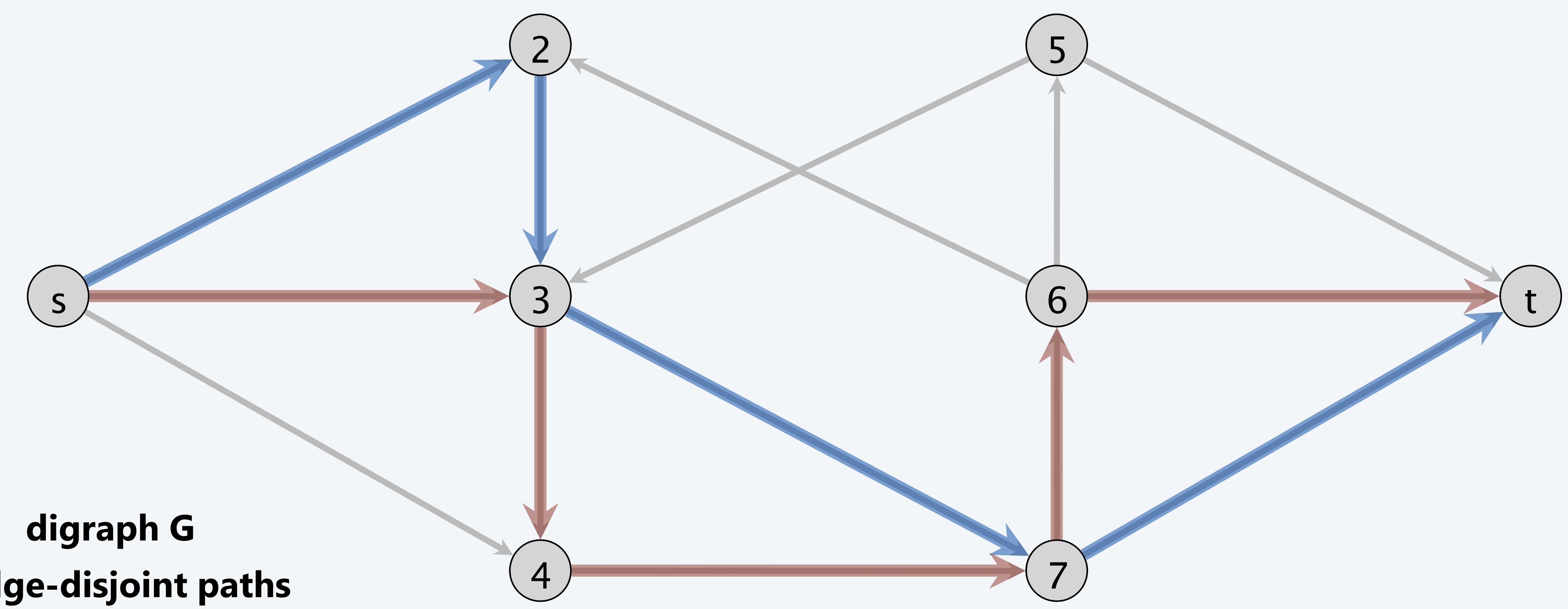
digraph G

Edge-disjoint paths

Def. Two paths are edge-disjoint if they have no edge in common.

Edge-disjoint paths problem. Given a digraph $G = (V, E)$ and two nodes s and t, find the max number of edge-disjoint $s \rightsquigarrow t$ paths.

Ex. Communication networks.



digraph G
2 edge-disjoint paths

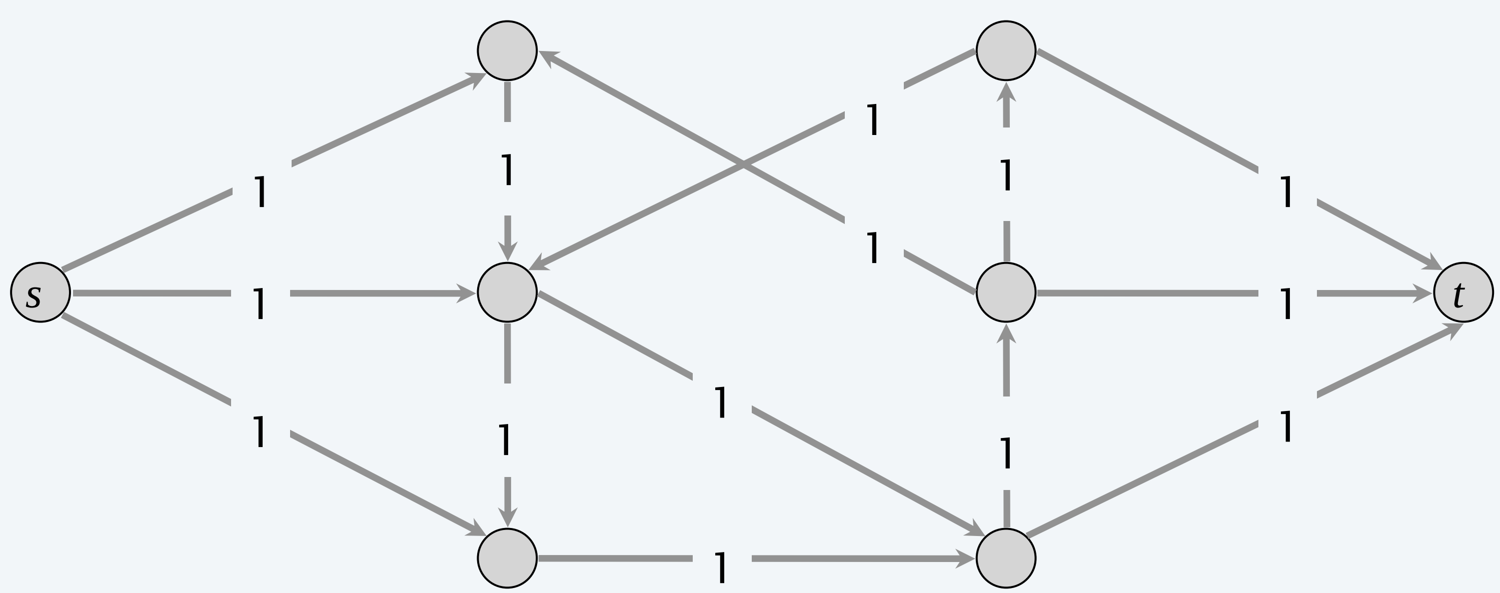
Edge-disjoint paths

Max-flow formulation. Assign unit capacity to every edge.

Theorem. 1–1 correspondence between k edge-disjoint $s \rightsquigarrow t$ paths in G and integral flows of value k in G'.

Pf. $\Rightarrow$

- Let $P_1, \ldots, P_k$ be k edge-disjoint $s \rightsquigarrow t$ paths in G.
- Set $f(e) = \begin{cases} 1 & \text{edge } e \text{ participates in some path } P_j \\ 0 & \text{otherwise} \end{cases}$
- Since paths are edge-disjoint, f is a flow of value k. ▪



Edge-disjoint paths

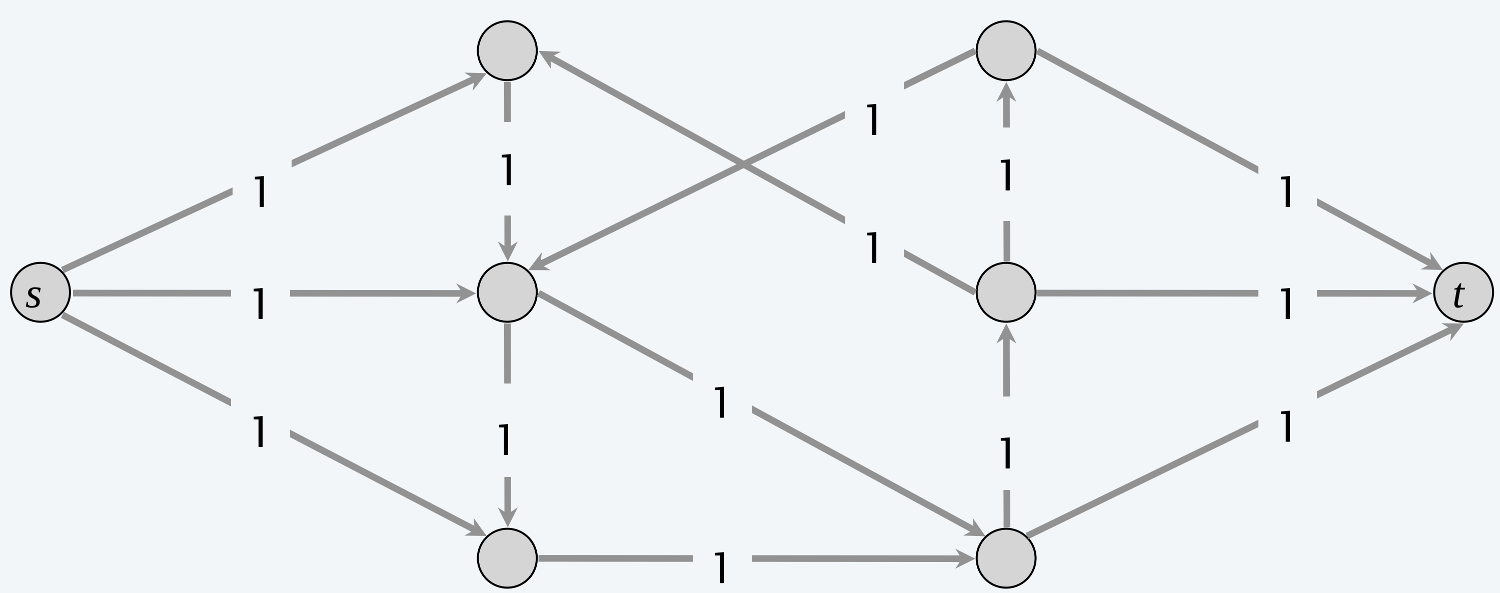
Max-flow formulation. Assign unit capacity to every edge.

Theorem. 1–1 correspondence between k edge-disjoint $s \rightsquigarrow t$ paths in G and integral flows of value k in G'.

Pf. $\Leftarrow$

- Let f be an integral flow in G' of value k.
- Consider edge (s, u) with $f(s, u) = 1$.
  - by flow conservation, there exists an edge (u, v) with $f(u, v) = 1$
  - continue until reach t, always choosing a new edge
- Produces k (not necessarily simple) edge-disjoint paths. ▪

can eliminate cycles to get simple paths in $O(mn)$ time if desired (flow decomposition)



Edge-disjoint paths

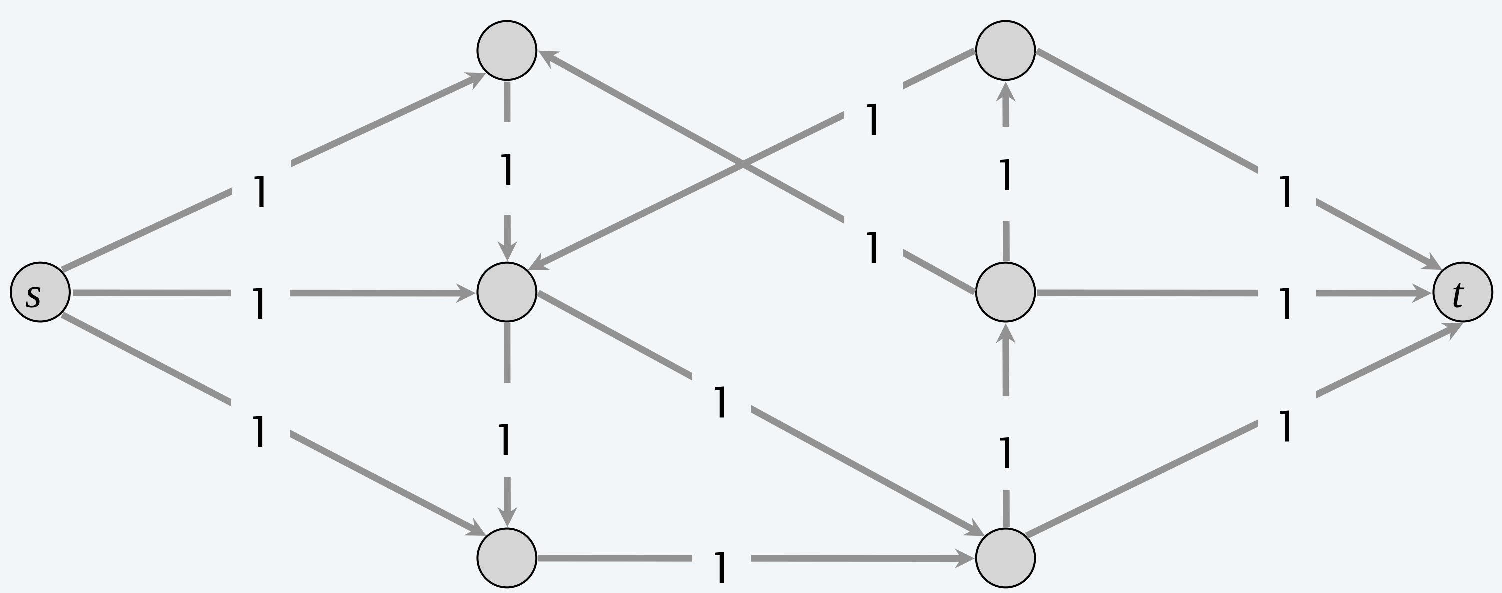
Max-flow formulation. Assign unit capacity to every edge.

Theorem. 1–1 correspondence between k edge-disjoint $s \rightsquigarrow t$ paths in G and integral flows of value k in G'.

Corollary. Can solve edge-disjoint paths problem via max-flow formulation.

Pf.

- Integrality theorem $\Rightarrow$ there exists a max flow f^* in G' that is integral.
- 1–1 correspondence $\Rightarrow$ f^* corresponds to max number of edge-disjoint $s \rightsquigarrow t$ paths in G. ▪



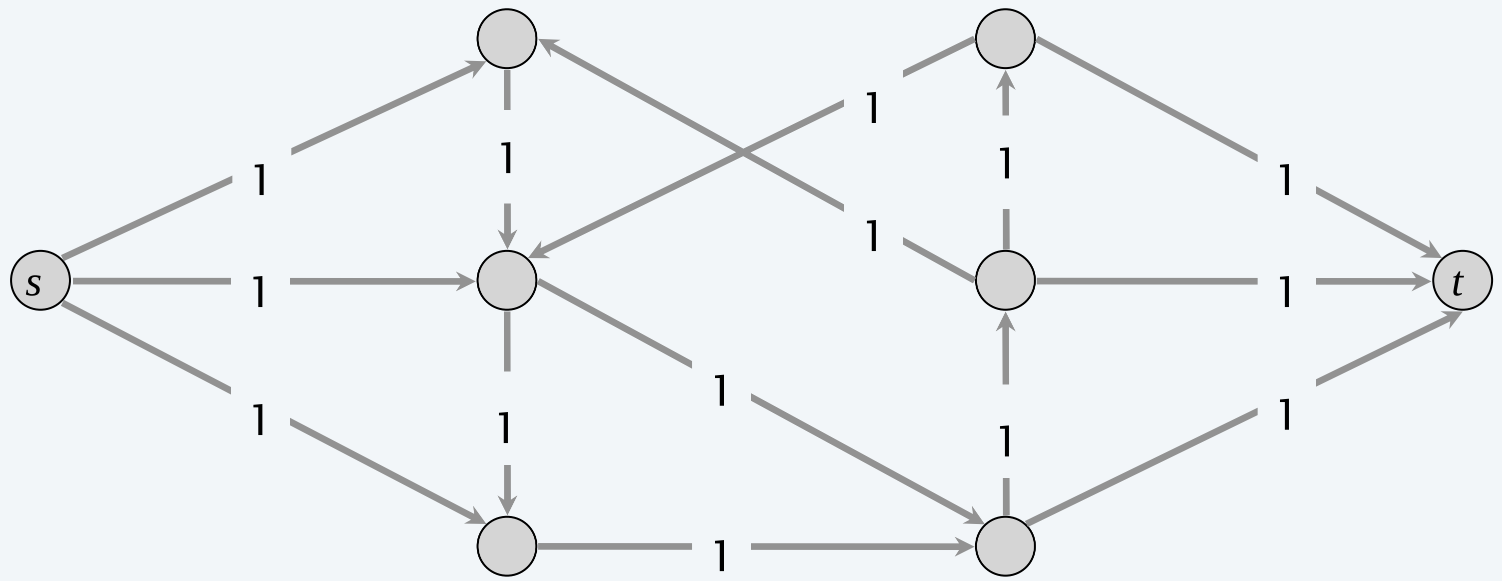
Edge-disjoint paths: running time

Max-flow formulation. Assign unit capacity to every edge.

Theorem. 1–1 correspondence between k edge-disjoint $s \leadsto t$ paths in G and integral flows of value k in G'.

Corollary. Can solve edge-disjoint paths problem via max-flow formulation.
Running time:

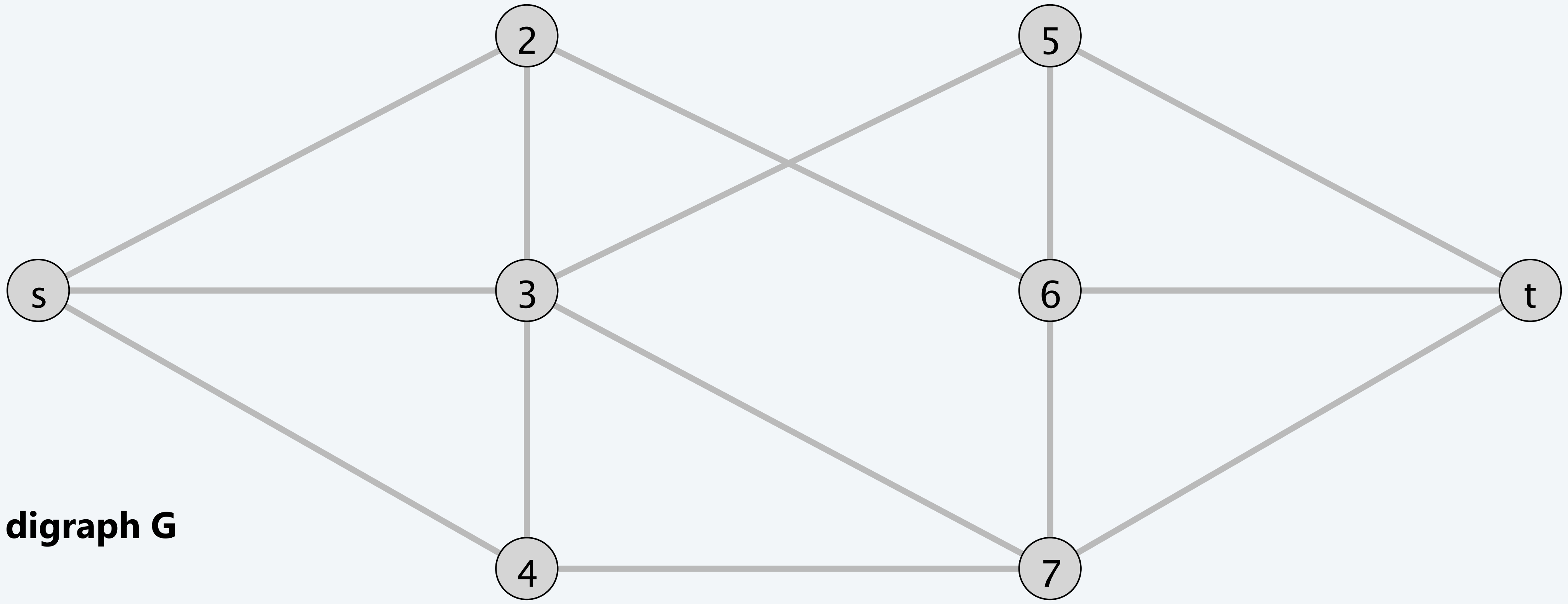
- Using Ford-Fulkerson:
- $\leq n$ augmentations $\Rightarrow O(mn)$ time.



Edge-disjoint paths in undirected graphs

Def. Two paths are edge-disjoint if they have no edge in common.

Edge-disjoint paths problem in undirected graphs. Given a graph $G = (V, E)$ and two nodes s and t, find the max number of edge-disjoint s–t paths.

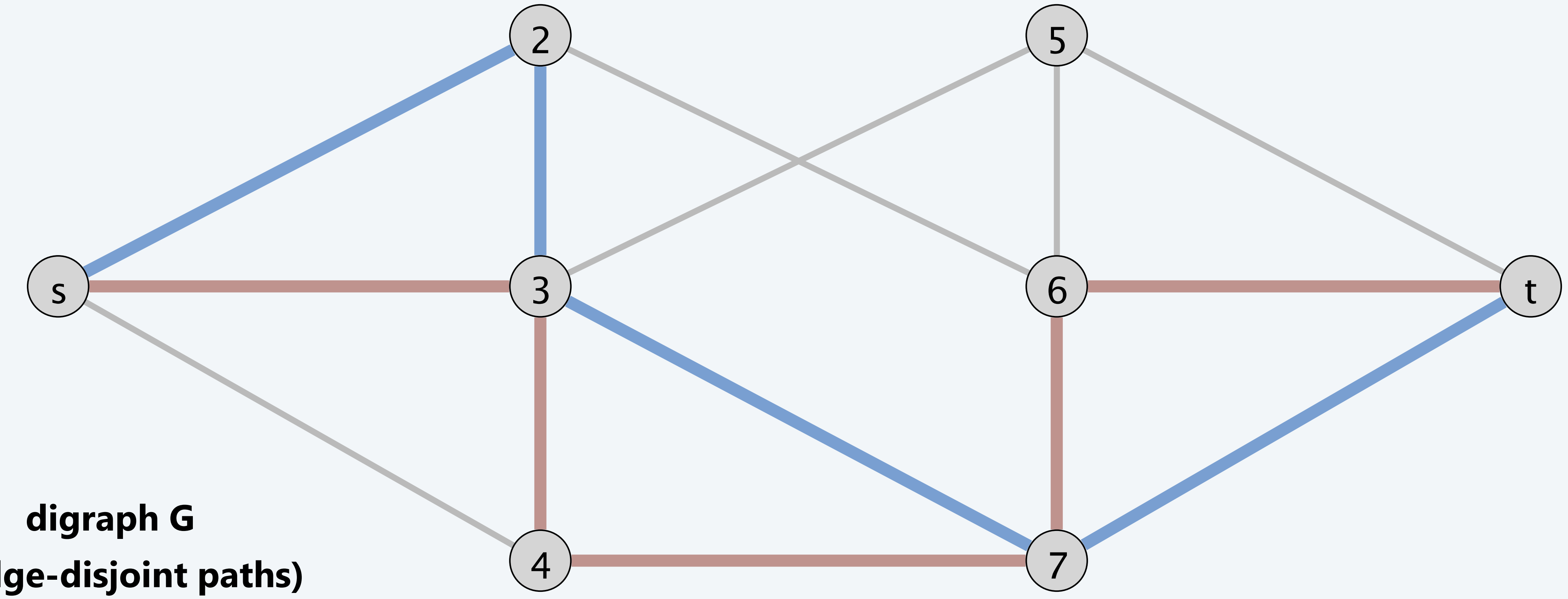


digraph G

Edge-disjoint paths in undirected graphs

Def. Two paths are edge-disjoint if they have no edge in common.

Edge-disjoint paths problem in undirected graphs. Given a graph $G = (V, E)$ and two nodes s and t, find the max number of edge-disjoint s–t paths.



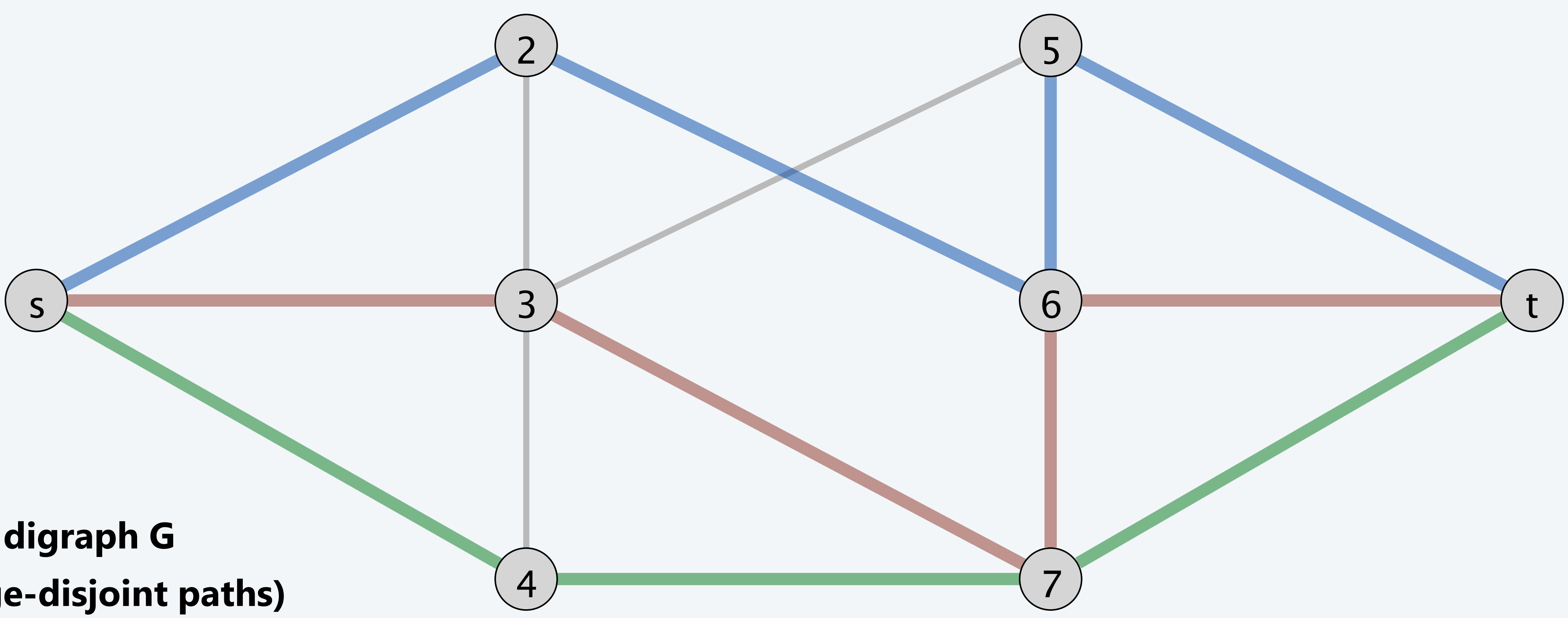
digraph G
(2 edge-disjoint paths)

Edge-disjoint paths in undirected graphs

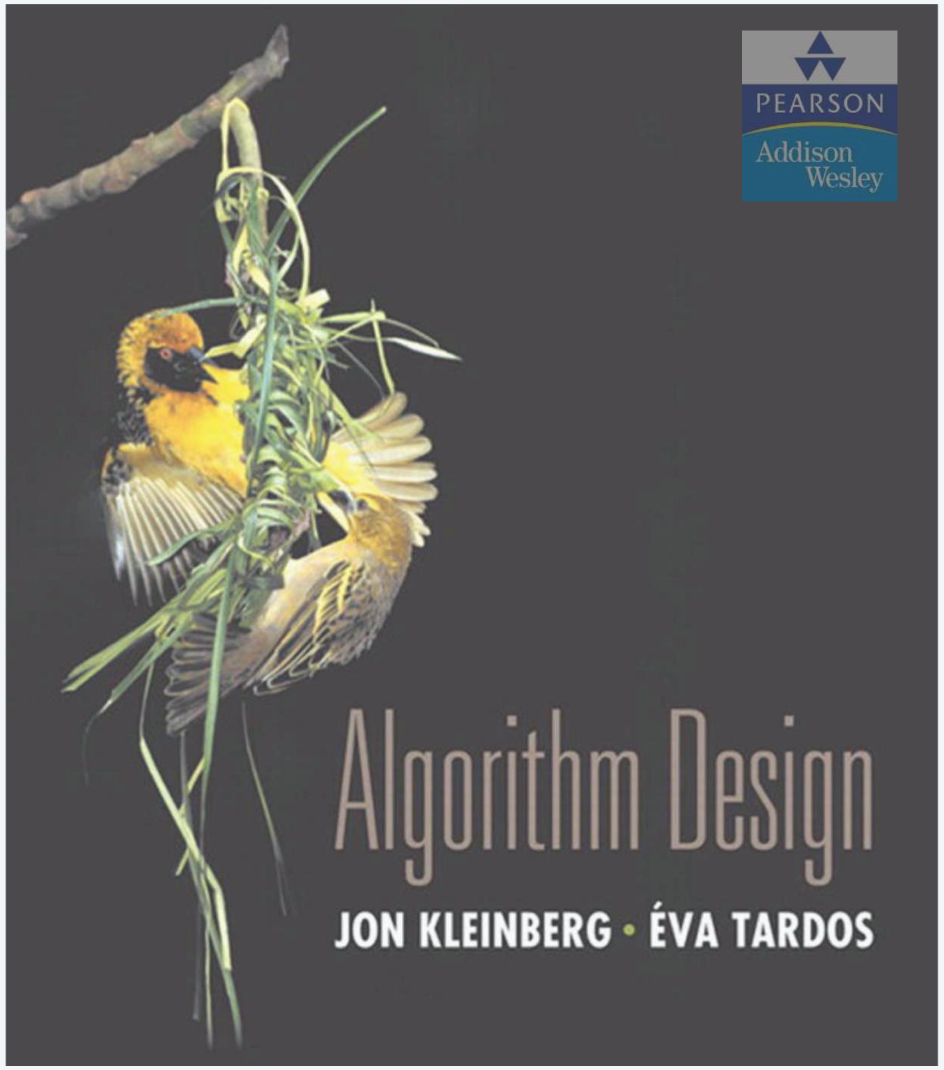
Def. Two paths are edge-disjoint if they have no edge in common.

Edge-disjoint paths problem in undirected graphs. Given a graph $G = (V, E)$ and two nodes s and t, find the max number of edge-disjoint s–t paths.

Exercise: design a max-flow-based algorithm for the problem.



digraph G
(3 edge-disjoint paths)



SECTION 7.10

7. NETWORK FLOW II

- *bipartite matching*
- *disjoint paths*
- *image segmentation*
- *baseball elimination*

Image segmentation

Image segmentation.

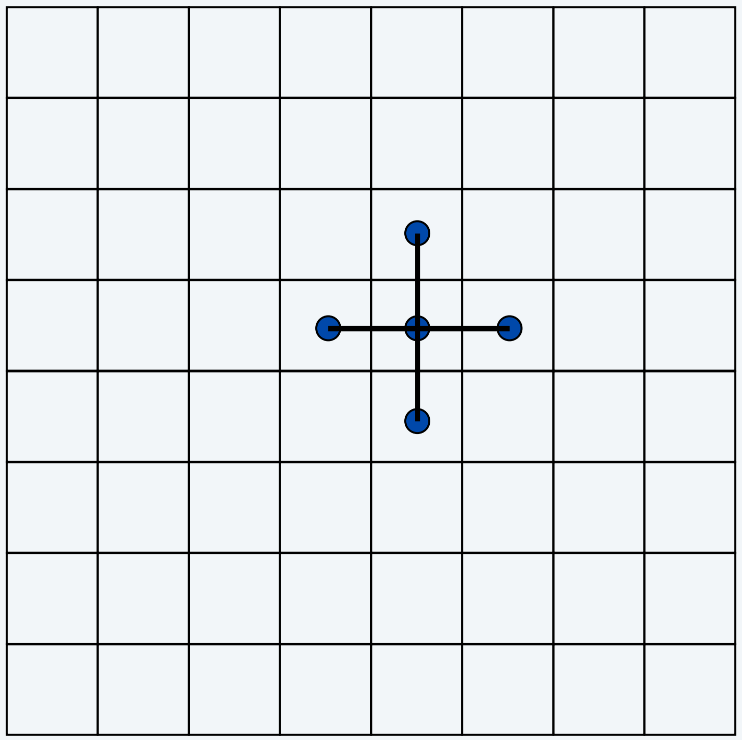
- Divide image into coherent regions.
- Central problem in image processing.

Ex. Separate human and robot from background scene.

Image segmentation

Foreground / background segmentation.

- Label each pixel in picture as belonging to foreground or background.
- V = set of pixels, E = pairs of neighboring pixels.
- $a_i \geq 0$ is likelihood pixel i in foreground.
- $b_i \geq 0$ is likelihood pixel i in background.
- $p_{ij} \geq 0$ is separation penalty for labeling one of i and j as foreground, and the other as background.



Goals.

- Accuracy: if $a_i > b_i$ in isolation, prefer to label i in foreground.
- Smoothness: if many neighbors of i are labeled foreground, we should be inclined to label i as foreground.
- Find partition (A, B) that maximizes: $\displaystyle \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$

foreground background

Image segmentation

Formulate as min-cut problem.

- Maximization.
- No source or sink.
- Undirected graph.

Turn into minimization problem.

- Maximizing $$\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$$

- is equivalent to minimizing

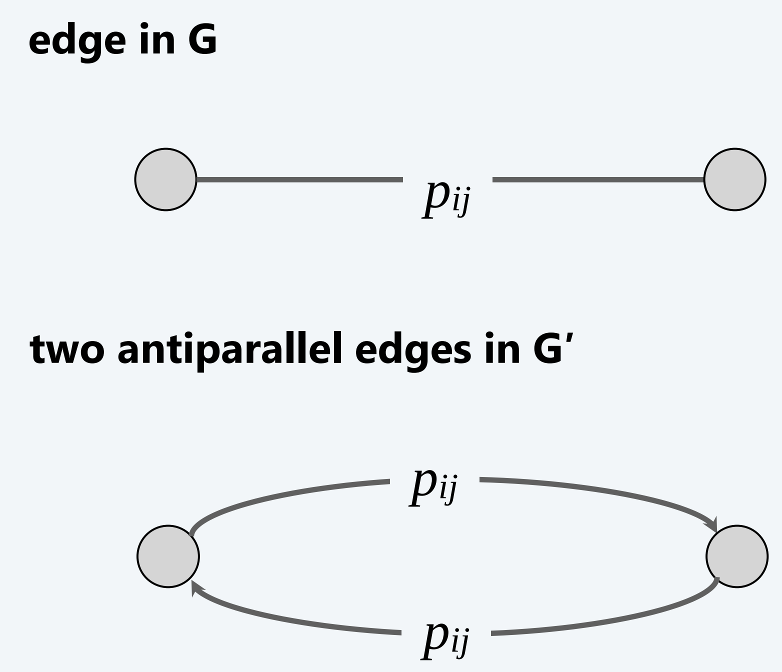
$$\underbrace{\left(\sum_{i \in V} a_i + \sum_{j \in V} b_j \right)}_{\text{a constant}} - \sum_{i \in A} a_i - \sum_{j \in B} b_j + \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$$

- or alternatively $$\sum_{j \in B} a_j + \sum_{i \in A} b_i + \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$$

Image segmentation

Formulate as min-cut problem $G' = (V', E')$.

- Include node for each pixel.
- Use two antiparallel edges instead of undirected edge.
- Add source s to correspond to foreground.
- Add sink t to correspond to background.



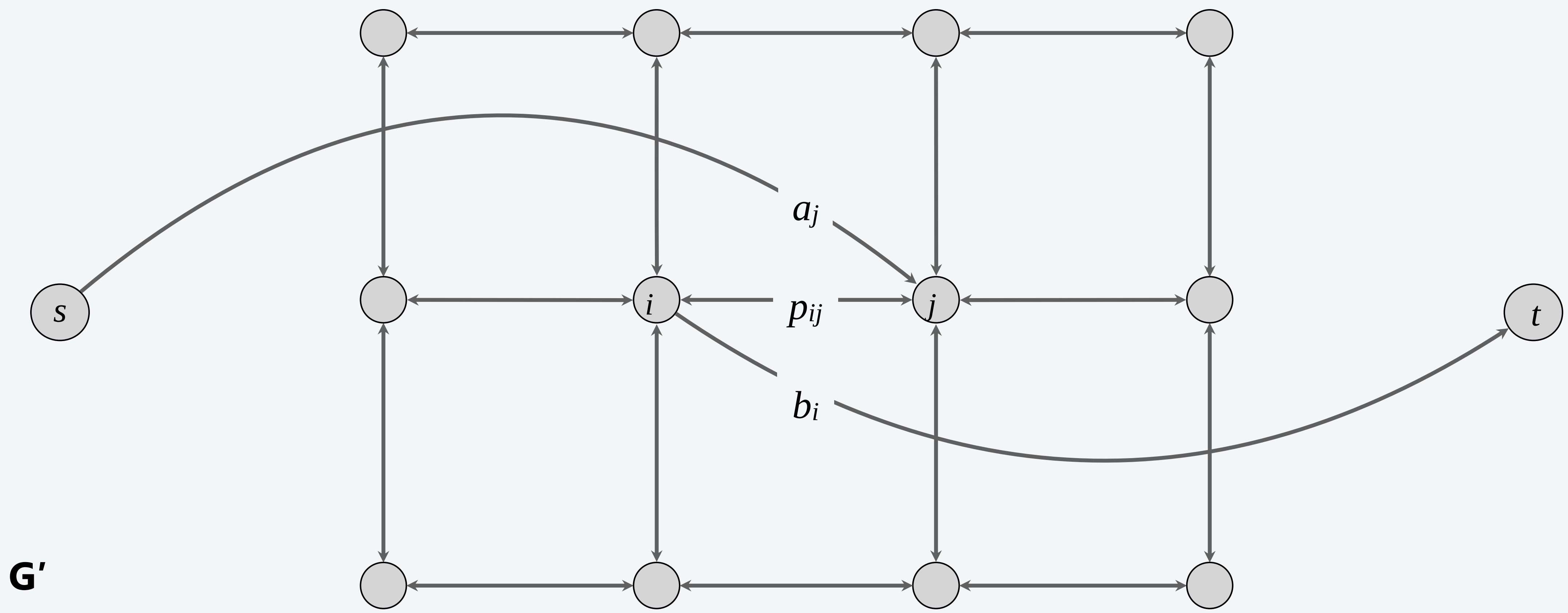


Image segmentation

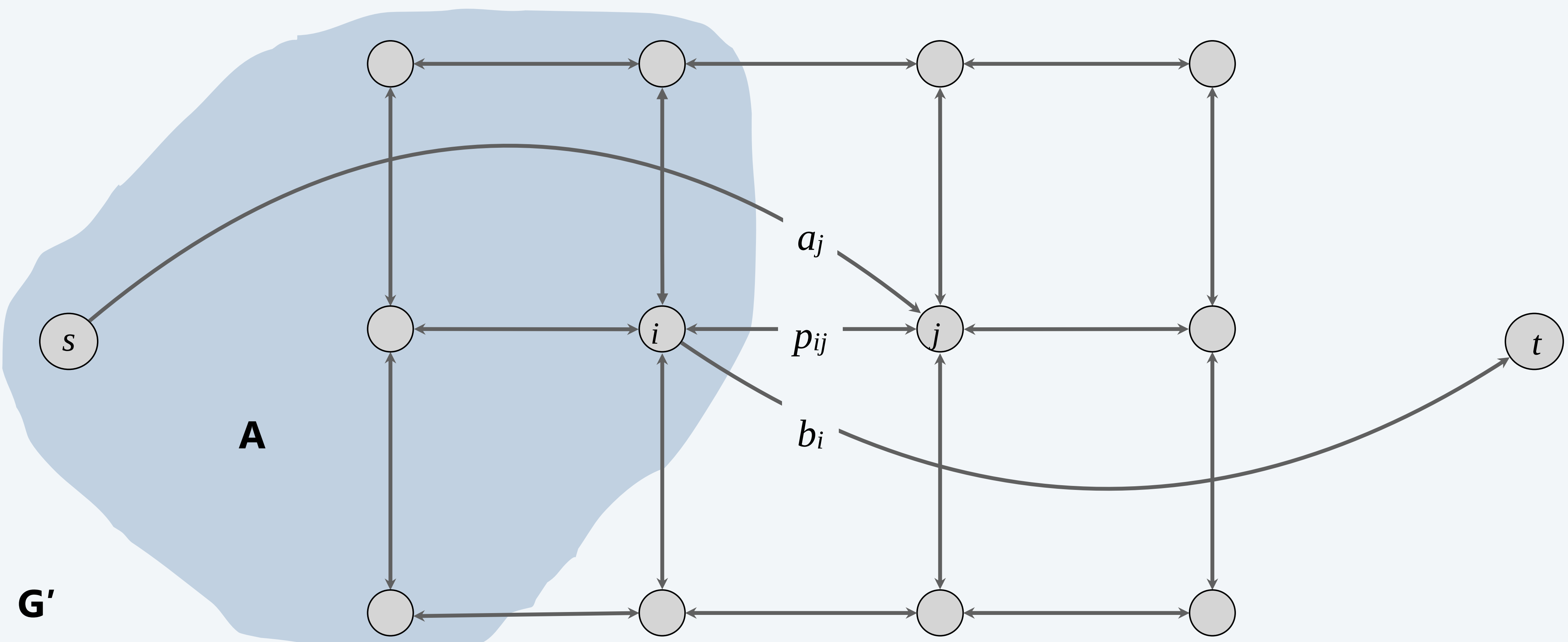
Consider min cut (A, B) in G'.

- A = foreground.

$$cap(A,B) \;=\; \sum_{j \in B} a_j \;+\; \sum_{i \in A} b_i \;+\; \sum_{\substack{(i,j) \in E \\ i \in A,\, j \in B}} p_{ij}$$

if i and j on different sides, p_{ij} counted exactly once

- Precisely the quantity we want to minimize.



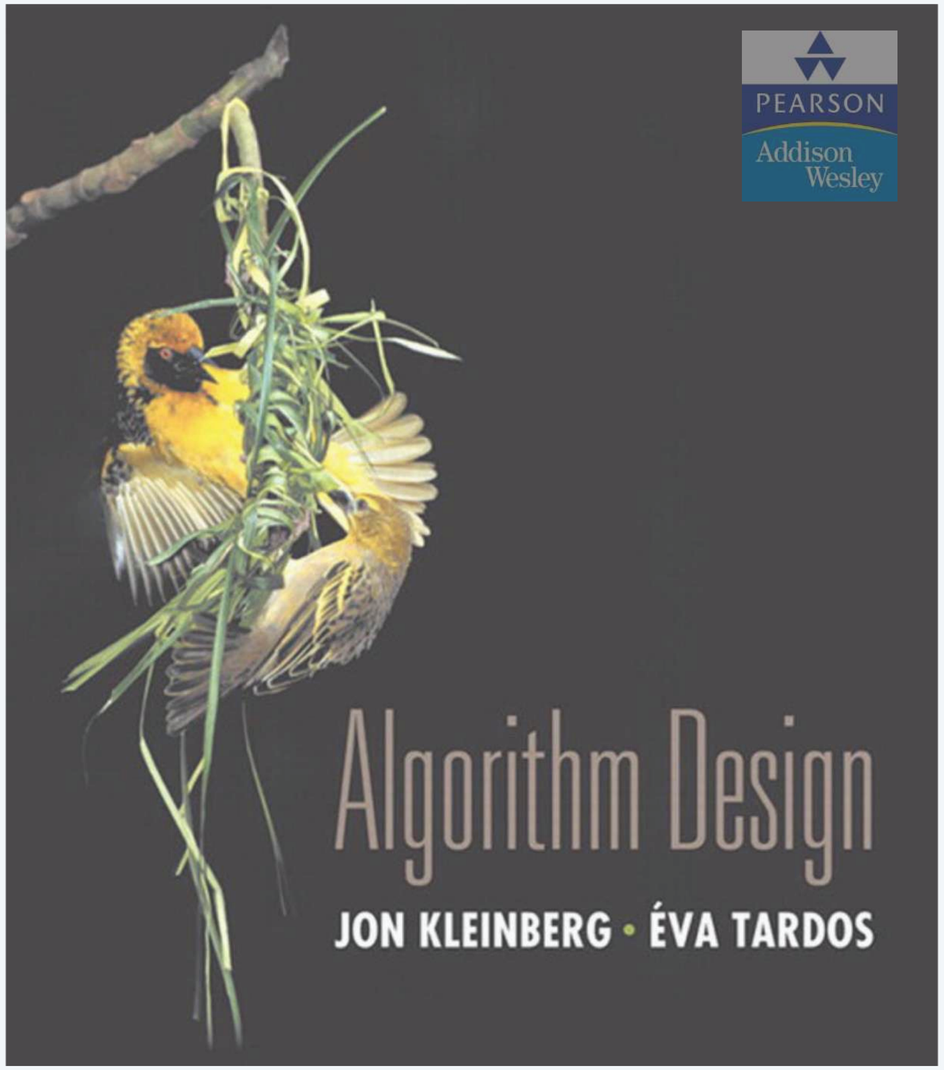
Grabcut image segmentation

Grabcut. [Rother–Kolmogorov–Blake 2004]

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts

Carsten Rother* Vladimir Kolmogorov† Andrew Blake‡
Microsoft Research Cambridge, UK

Figure 1: **Three examples of GrabCut .** The user drags a rectangle loosely around an object. The object is then extracted automatically.



SECTION 7.12

7. NETWORK FLOW II

- *bipartite matching*
- *disjoint paths*
- *image segmentation*
- *baseball elimination*

Baseball elimination

TUESDAY, SEPTEMBER 10, 1996

San Francisco Chronicle

The Gate

Sports Online

▶ http://www.sfgate.com

SPORTING G

49ers, Young Get Big Bre

Quarterback m

By Gary Swan

Chronicle Staff Writer

The bye week has come at a perfect time for the 49ers and quarterback Steve Young. If they had a game next Sunday, there's a good chance Young would not play.

But the pulled groin muscle on his up-

Giants Officially Leave the NL West Race

By Nancy Gay

Chronicle Staff Writer

CARDINALS 6
GIANTS 2

With the smack of another National League West bat 500 miles away, the Giants' run at the division title ended last night, just as they were handing the visiting St. Louis Cardinals an even bigger lead in the NL Central.

In San Diego, Greg Vaughn's three-run homer in the eighth pushed the Padres over the Pirates and officially shoved the rest of the Giants' season into the background. On the heels of their tedious 6-2 loss before an announced crowd of 10,307 at Candlestick Park, the Giants fell 19½ games off the lead.

As it is, the worst the Padres (80-65) can finish is 80-82. The Giants have fallen to 59-83 with 20

Financing in Place For Giants' New Stadium

SEE PAGE B1, MAIN NEWS

games left; they cannot win 80 games. Coming off a miserable 2-8 mark on a three-city road trip that saw their road record drop to 27-47, the Giants were hoping to get off on the right foot in their longest homestand of the year (15 games, 14 days).

"Where we are, you're going to be eliminated sooner or later," Baker said quietly. "But it doesn't alter the fact that we've still got to play ball. You've still got to play hard, the fans come out to watch you play. You've got to play for the fact of loving to play, no matter where you are in the standings.

"You've got to play the role of spoiler, to not make it easier on

GIANTS: *Page D5 Col. 3*

Baseball elimination problem

Q. Which teams have a chance of finishing the season with the most wins?

i	team		wins	losses	to play	ATL	PHI	NYM	MON
0		Atlanta	83	71	8	–	1	6	1
1		Philly	80	79	3	1	–	0	2
2		New York	78	78	6	6	0	–	0
3		Montreal	77	82	3	1	2	0	–

Montreal is mathematically eliminated.

- Montreal finishes with ≤ 80 wins.
- Atlanta already has 83 wins.

Remark. This is the only reason sports writers appear to be aware of — conditions are sufficient but not necessary!

Baseball elimination problem

Q. Which teams have a chance of finishing the season with the most wins?

i	team		wins	losses	to play	ATL	PHI	NYM	MON
0		Atlanta	83	71	8	–	1	6	1
1		Philly	80	79	3	1	–	0	2
2		New York	78	78	6	6	0	–	0
3		Montreal	77	82	3	1	2	0	–

Philadelphia is mathematically eliminated.

- Philadelphia finishes with ≤ 83 wins.
- Either New York or Atlanta will finish with ≥ 84 wins.

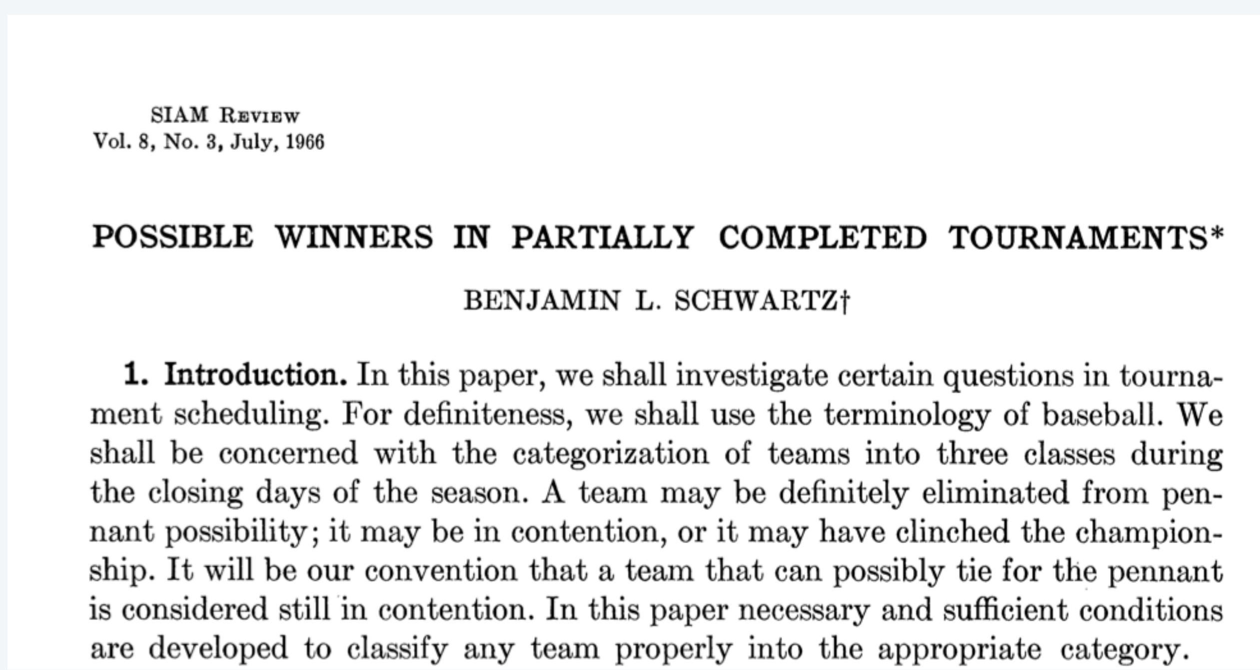
Observation. Answer depends not only on how many games already won and left to play, but on whom they're against.

Baseball elimination problem

Current standings.

- Set of teams S.
- Distinguished team $z \in S$.
- Team x has won w_x games already.
- Teams x and y play each other r_{xy} additional times.

Baseball elimination problem. Given the current standings, is there any outcome of the remaining games in which team z finishes with the most (or tied for the most) wins?



SIAM Review
Vol. 8, No. 3, July, 1966

POSSIBLE WINNERS IN PARTIALLY COMPLETED TOURNAMENTS*

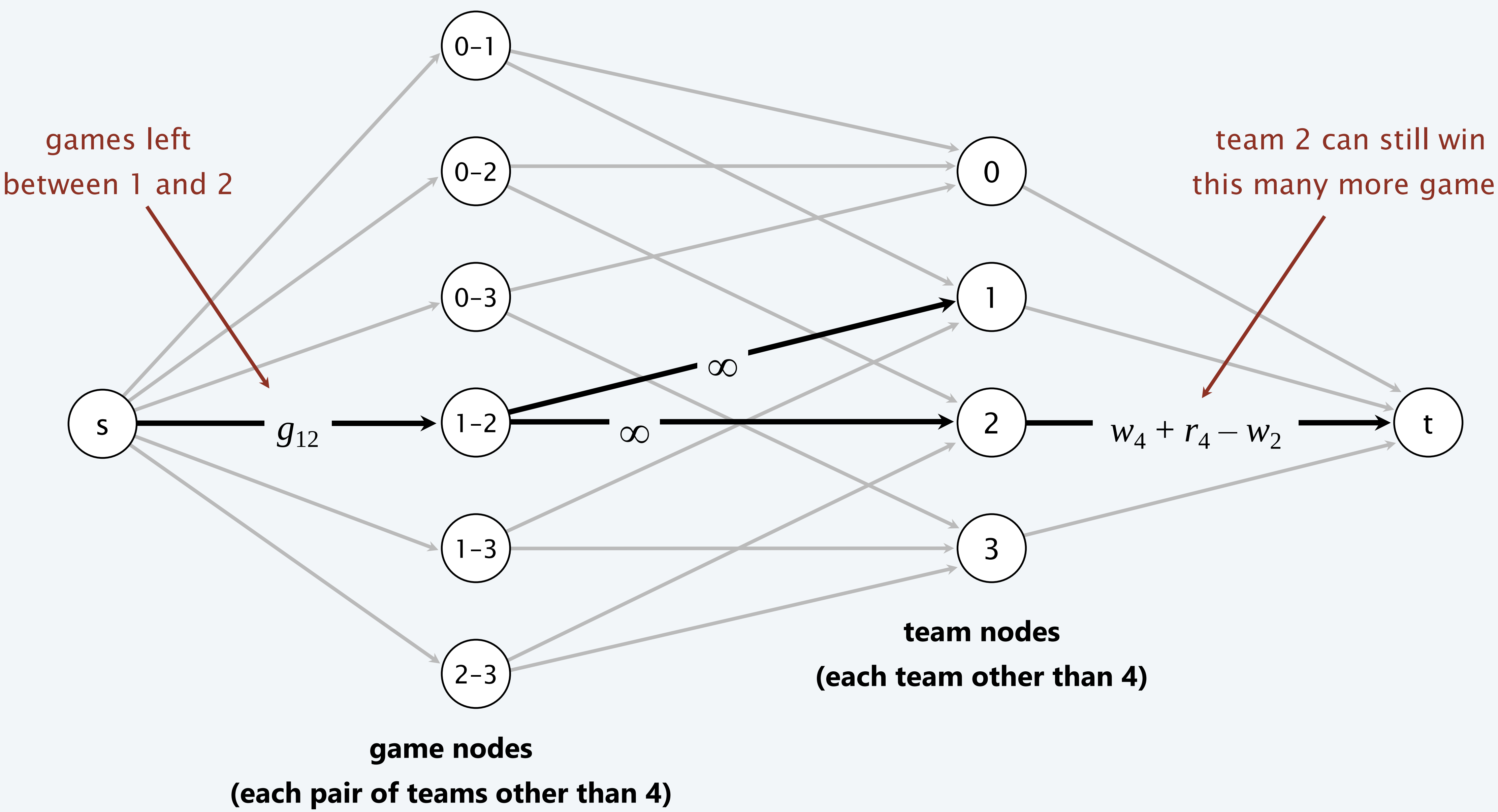
BENJAMIN L. SCHWARTZ†

1. Introduction. In this paper, we shall investigate certain questions in tournament scheduling. For definiteness, we shall use the terminology of baseball. We shall be concerned with the categorization of teams into three classes during the closing days of the season. A team may be definitely eliminated from pennant possibility; it may be in contention, or it may have clinched the championship. It will be our convention that a team that can possibly tie for the pennant is considered still in contention. In this paper necessary and sufficient conditions are developed to classify any team properly into the appropriate category.

Baseball elimination problem: max-flow formulation

Can team 4 finish with most wins?

- W.l.o.g. assume team 4 wins all remaining games $\Rightarrow$ $w_4 + r_4$ wins.
- Divvy remaining games so that all teams have $\leq w_4 + r_4$ wins.



Baseball elimination problem: max-flow formulation

Theorem. Team 4 not eliminated iff max flow saturates all edges leaving s.

Pf.

- Integrality theorem $\Rightarrow$ each remaining game between x and y added to number of wins for team x or team y.
- Capacity on (x, t) edges ensure no team wins too many games. ▪