Algorithm Design and Analysis

CSE 565

LECTURE 27

Network Flow

- Application:
  Bipartite Matching

Adam Smith

Recently: Ford-Fulkerson

- Find max s-t flow & min s-t cut in $O(mnC)$ time
  - All capacities are integers $\leq$ C
  - (We will discuss how to remove this assumption)

- **Duality**: Max flow value = min cut capacity

- **Integrality**: if capacities are integers, then FF algorithm produces an integral max flow

Last time

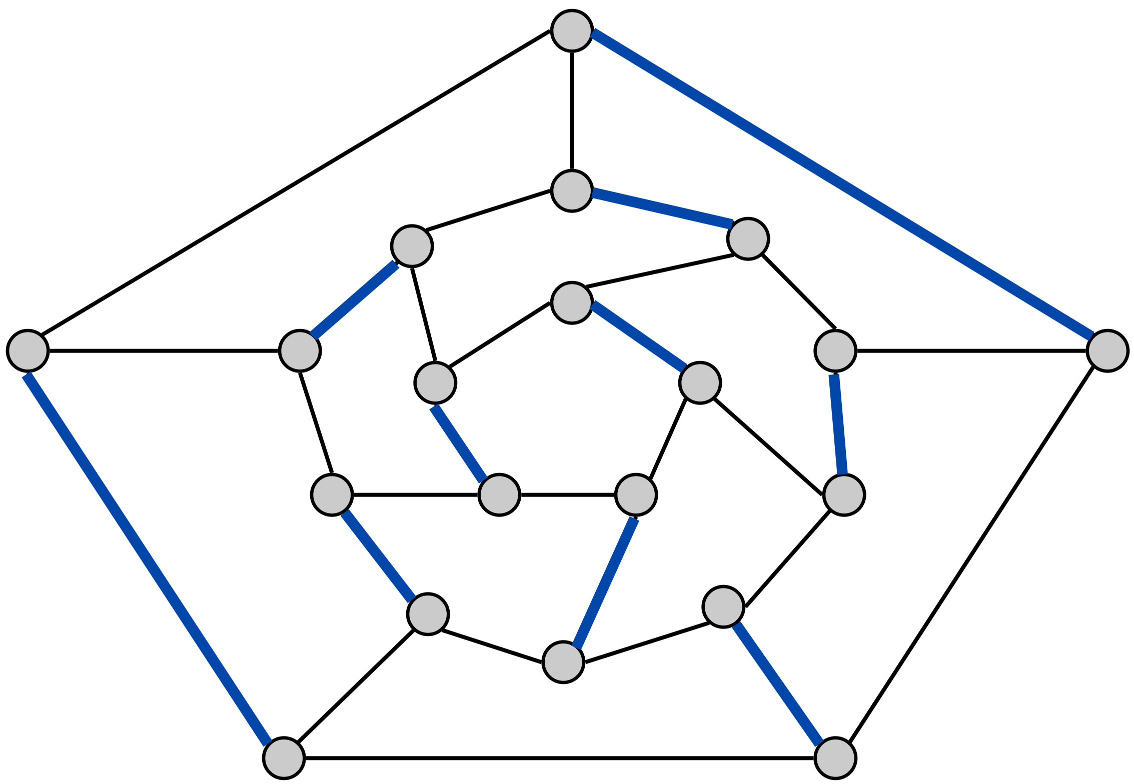
- Reductions
- Project Selection Problem (Section 7.11)
  – Reduction of project selection to min-cut

- Today:
  – Maximum bipartite matching
  – Reducing MBM to max-flow
  – Hall's theorem

Matching

Matching.

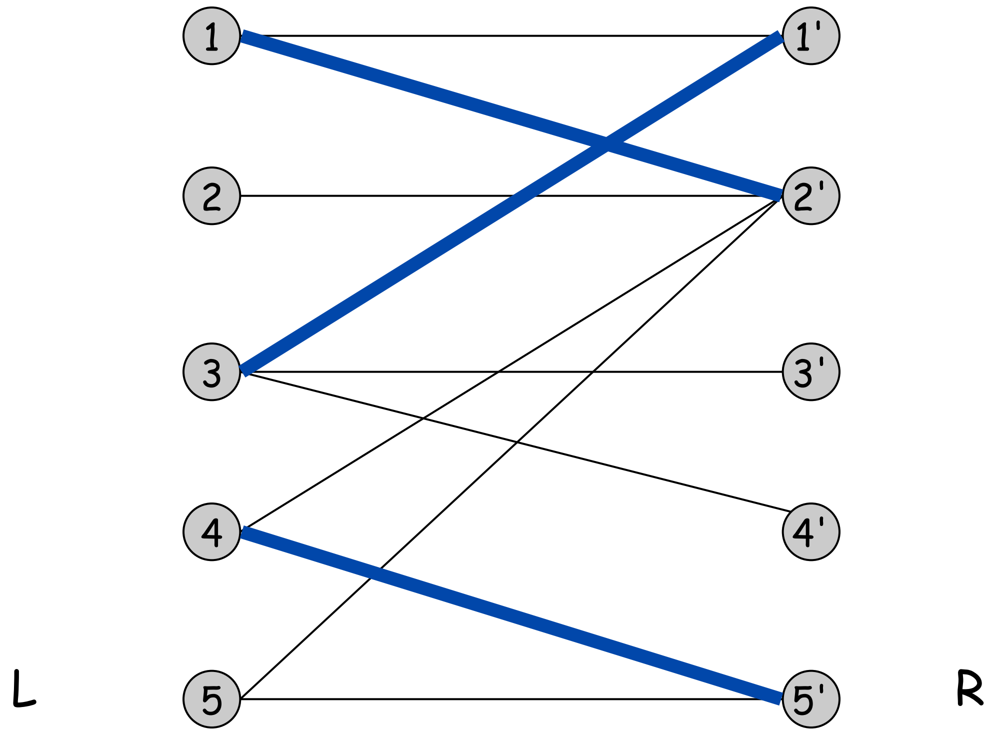
- Input: undirected graph G = (V, E).
- $M \subseteq E$ is a matching if each node appears in at most edge in M.
- Max matching: find a max cardinality matching.



Bipartite Matching

Bipartite matching.

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



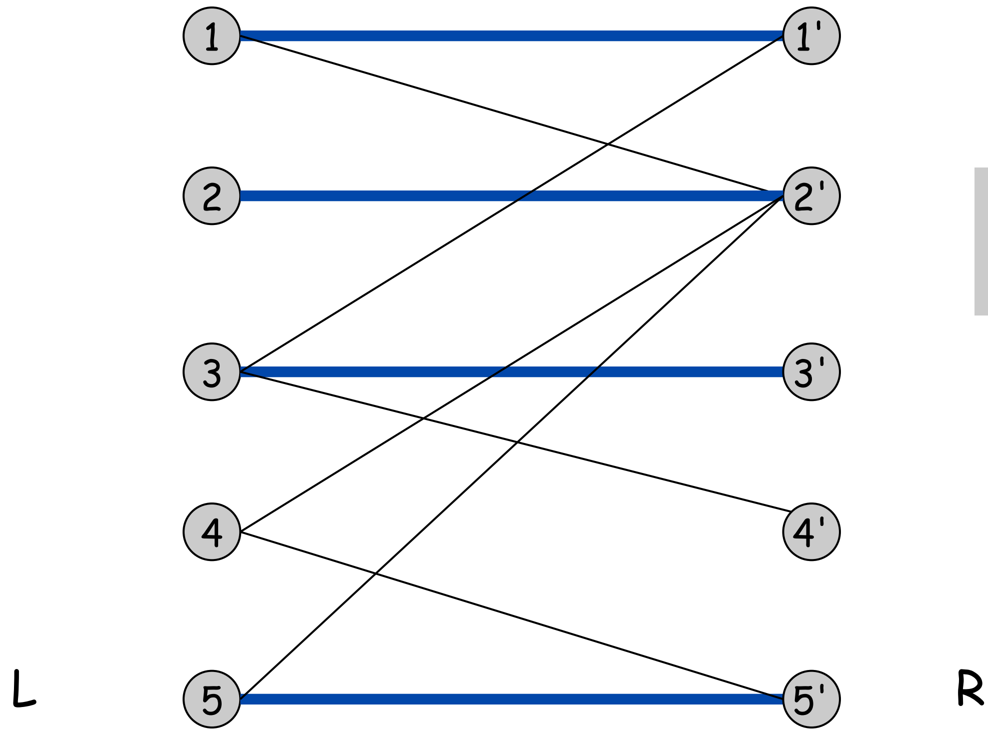
matching

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 edge in M.
- Max matching: find a max cardinality matching.



max matching

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

Reductions

Roughly: Problem A reduces to problem B if there is a simple algorithm for A that uses an algorithm for problem B as a subroutine.

Usually:

- Given instance x of problem A
  we find a instance x′ of problem B
- Solve x′
- Use the solution to build a solution to x

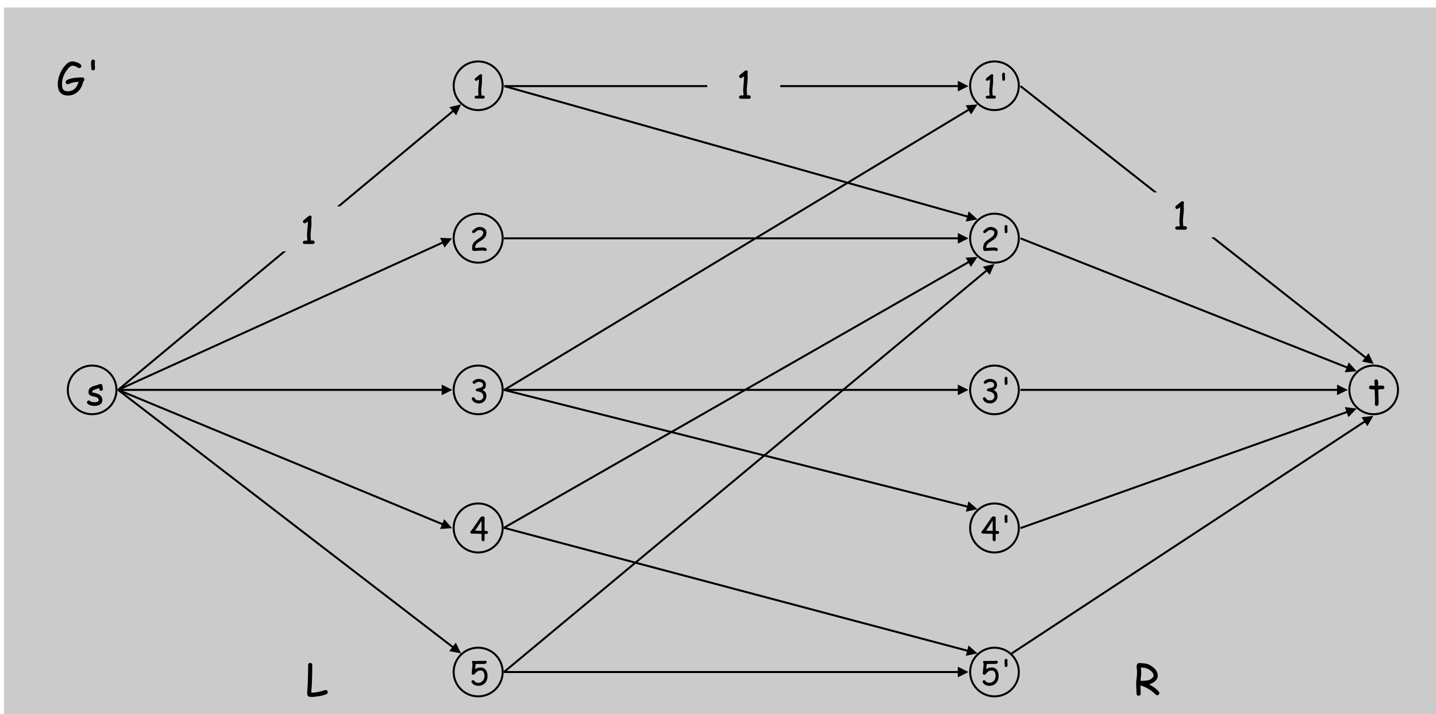
Useful skill: quickly identifying problems where existing solutions may be applied.

- Good programmers do this all the time

Reducing Bipartite Matching to Maximum Flow

Reduction to Max flow.

- Create digraph $G' = (L \cup R \cup \{s, t\}, E')$.
- 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

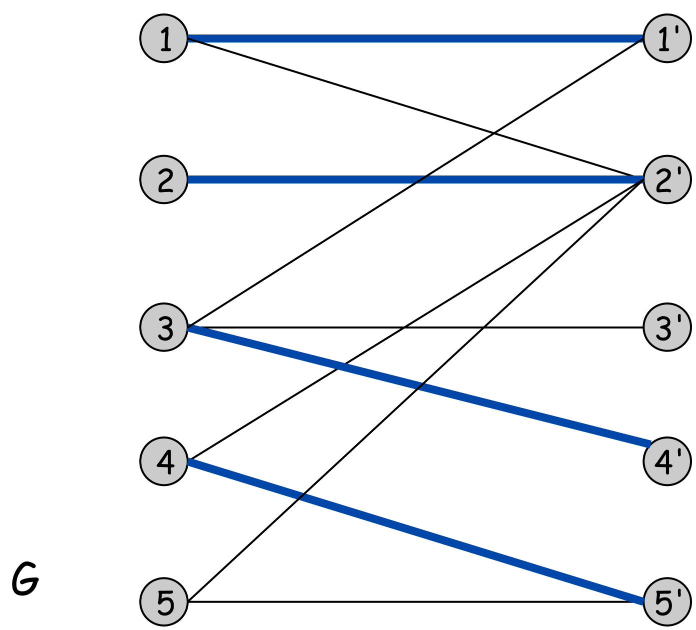
- max. matching in $G \leq$ max flow in G'
- max. matching in $G \geq$ max flow in G'

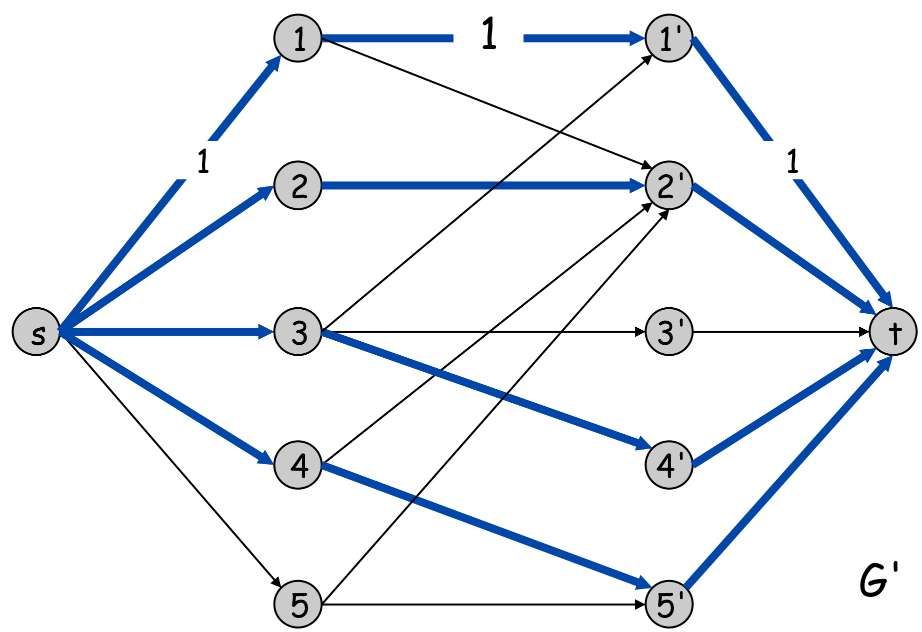
Bipartite Matching: Proof of Correctness

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

Pf. ≤

- Given max matching M of cardinality k.
- Consider flow f that sends 1 unit along each of k paths.
- f is a flow, and has cardinality k. ▪



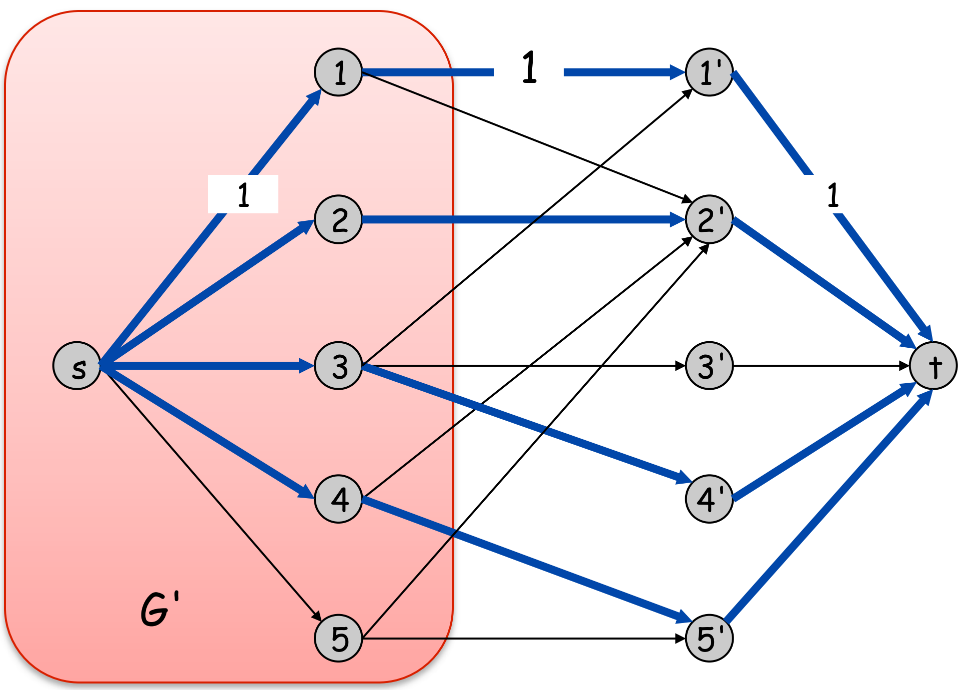


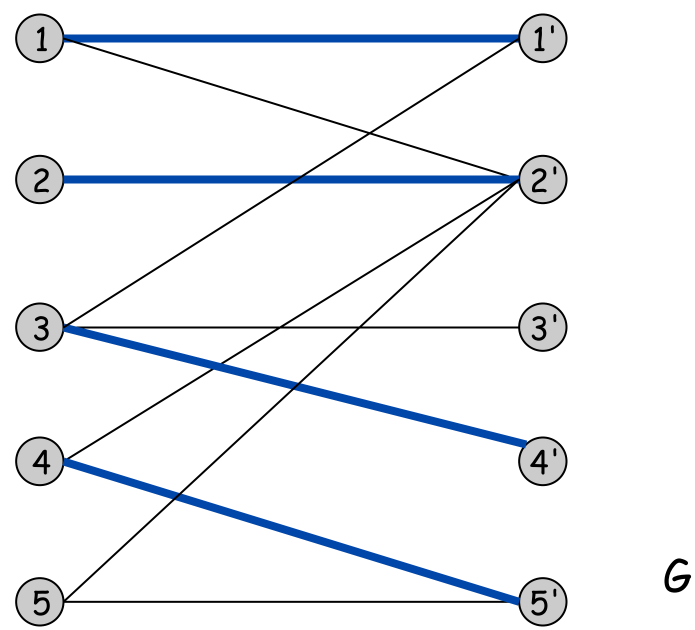
Bipartite Matching: Proof of Correctness

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

Pf. ≥

- Let f be a max flow in G' of value k.
- Integrality theorem ⇒ k is integral; all capacities are 1 ⇒ f is 0-1.
- 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
  - |M| = k: consider cut (L ∪ s, R ∪ t) ▪





Exercises

- Give an example where the greedy algorithm for MBM fails.
- How bad can the greedy algorithm be, i.e. how far can the size of the maximum matching (global max) be from the size of the greedy matching (local max)?
- What do augmenting paths look like in this max-flow instance?

Perfect Matching

Def. A matching $M \subseteq E$ is perfect if each node appears in exactly one edge in M.

Q. When does a bipartite graph have a perfect matching?

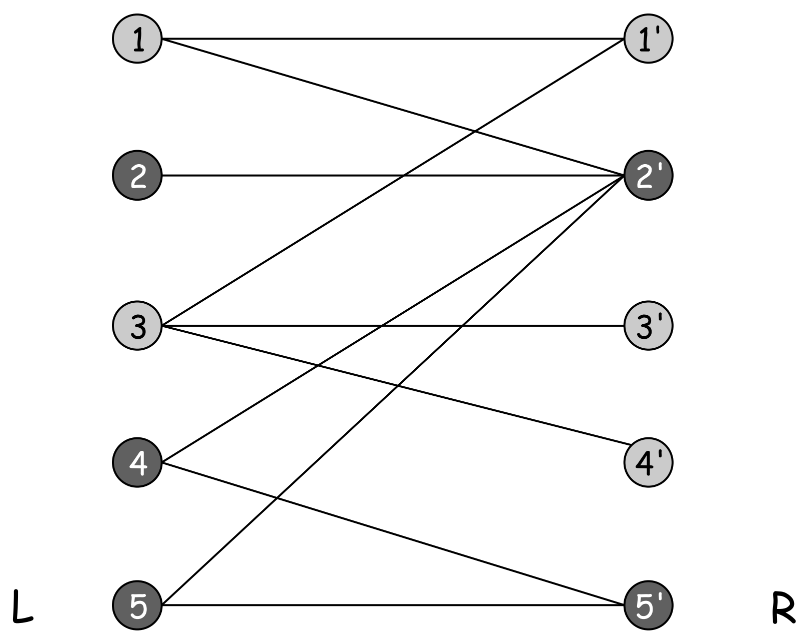
Structure of bipartite graphs with perfect matchings.

- Clearly we must have $|L| = |R|$.
- What other conditions are necessary?
- What conditions are sufficient?

Perfect Matching

Notation. Let S be a subset of nodes, and let N(S) be the set of nodes adjacent to nodes in S.

Observation. If a bipartite graph $G = (L \cup R, E)$, has a perfect matching, then $|N(S)| \geq |S|$ for all subsets $S \subseteq L$.
Pf. Each node in S has to be matched to a different node in N(S).



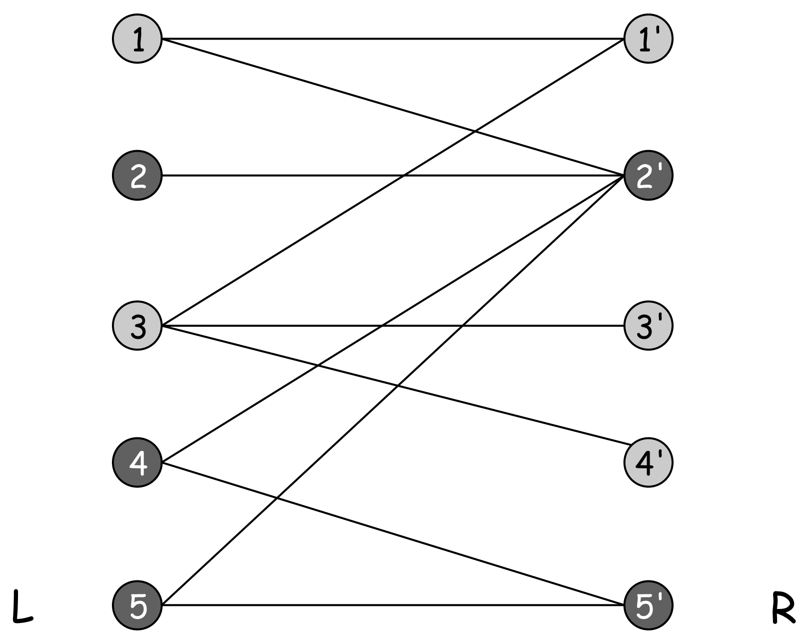
No perfect matching:
S = { 2, 4, 5 }
N(S) = { 2', 5' }.

Marriage Theorem

Marriage Theorem. [Frobenius 1917, Hall 1935] Let $G = (L \cup R, E)$ be a bipartite graph with $|L| = |R|$. Then, G has a perfect matching iff

$$|N(S)| \geq |S| \text{ for all subsets } S \subseteq L.$$

Pf. $\Rightarrow$ This was the previous observation.



No perfect matching:

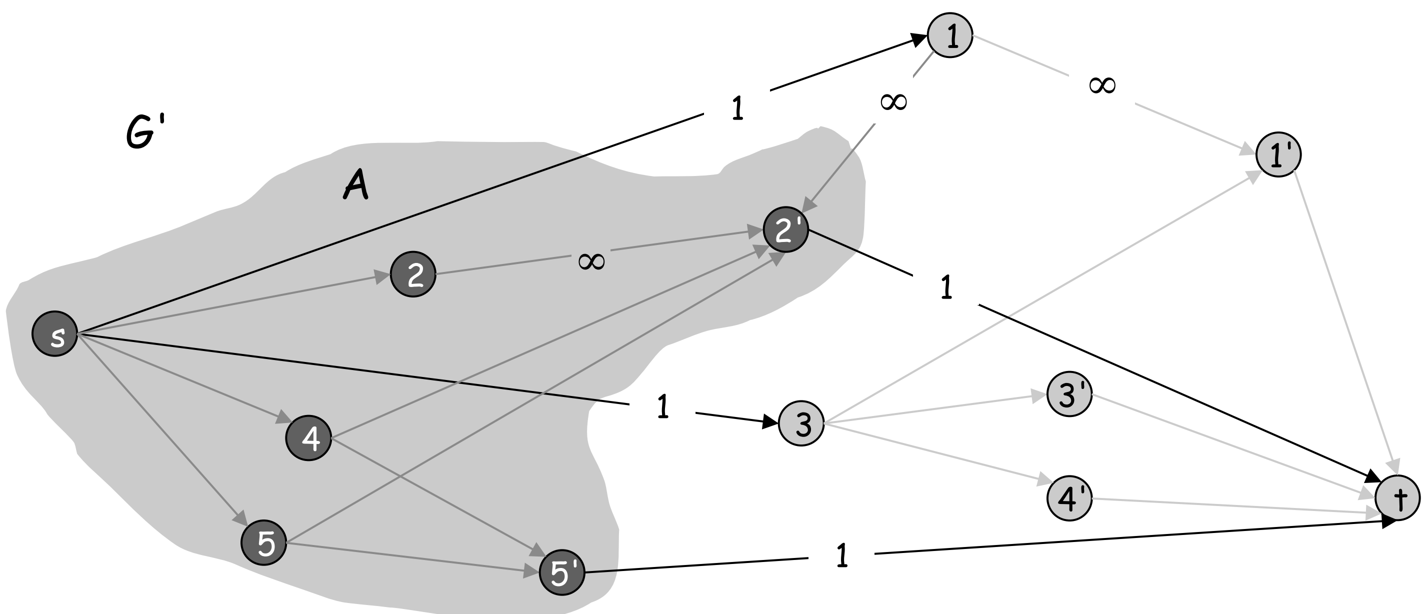
$S = \{ 2, 4, 5 \}$

$N(S) = \{ 2', 5' \}$.

Proof of Marriage Theorem

Pf. $\Leftarrow$ Suppose G does not have a perfect matching.

- Formulate as a max flow problem with ∞ constraints on edges from L to R and let (A, B) be min cut in G'.
- By max-flow min-cut, cap(A, B) < | L |.
- Choose $S = L \cap A$.
- $cap(A, B) = |L \cap B| + |R \cap A|$.
- Since min cut can't use ∞ edges: $N(S) \subseteq R \cap A$.
- $|N(S)| \le |R \cap A| = cap(A, B) - |L \cap B| < |L| - |L \cap B| = |S|$. ▪



S = {2, 4, 5}

L ∩ B = {1, 3}

R ∩ A = {2', 5'}

N(S) = {2', 5'}

Bipartite Matching: Running Time

Which max flow algorithm to use for bipartite matching?

- Generic augmenting path: $O(m \text{ val}(f^*)) = O(mn)$.
- Capacity scaling: $O(m^2 \log C) = O(m^2)$.
- Shortest augmenting path (not covered in class): $O(m\, n^{1/2})$.

Non-bipartite matching.

- Structure of non-bipartite graphs is more complicated, but well-understood. [Tutte-Berge, Edmonds-Galai]
- Blossom algorithm: $O(n^4)$. [Edmonds 1965]
- Best known: $O(m\, n^{1/2})$. [Micali-Vazirani 1980]
- Recently: better algorithms for dense graphs, e.g. $O(n^{2.38})$ [Harvey]