Algorithm Design and Analysis

CSE 565

LECTURES 19

Maximum Flow Applications

- With unit capacities
  - Bipartite matching
  - Perfect matching
  - Edge-disjoint paths

Sofya Raskhodnikova

Reminders: Max Flow Algorithms

- Algorithms to find max *s*-*t* flow & min *s*-*t* cut when capacities are integers $\leq C$
  - Ford-Fulkerson runs in $\boldsymbol{O(nmC)}$ time
  - the scaling max-flow algorithm runs in $\boldsymbol{O(m^2 \log C)}$ time.

- **Duality**: Max flow value = min cut capacity
- **Integrality**: If capacities are integers, then both algorithms produce an integral max flow

Review Question

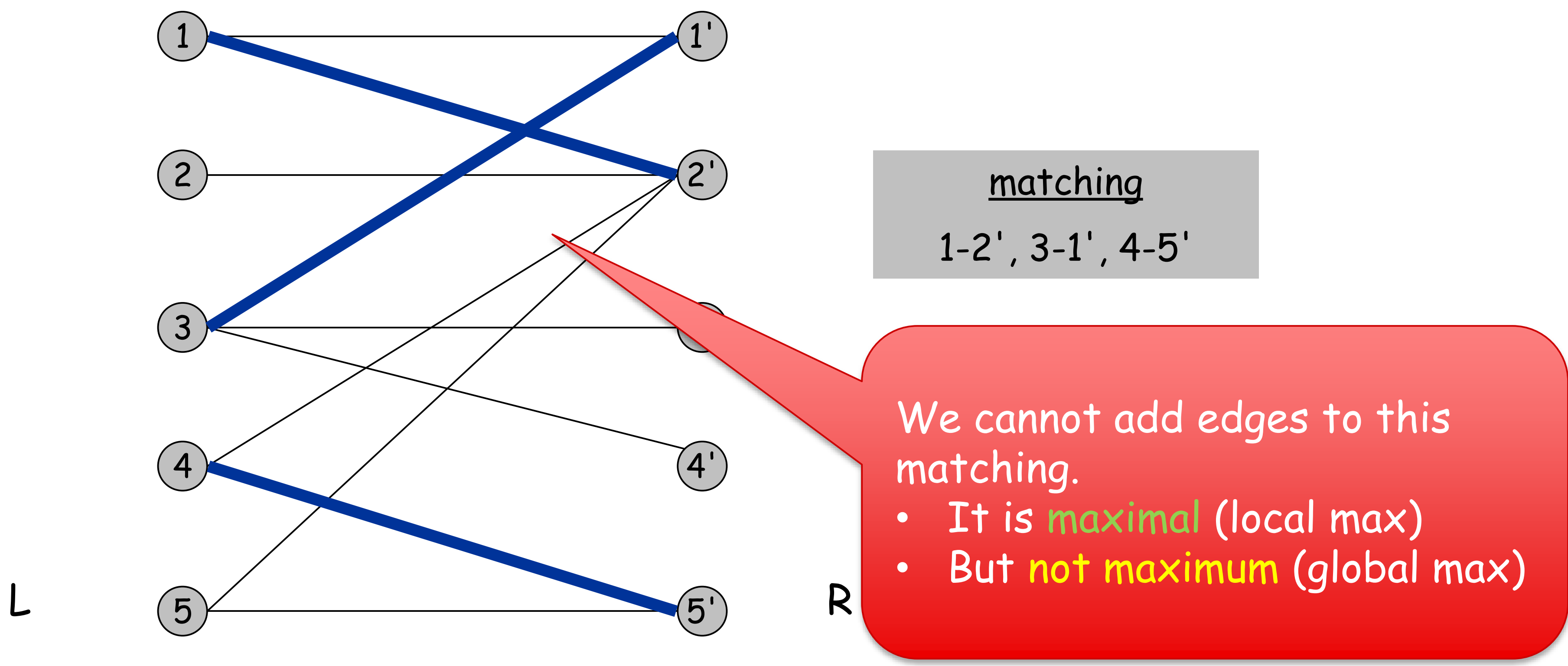
Suppose we run Ford Fulkerson in a graph where all capacities are in $\{0,1,\ldots,C,\infty\}$, but the value of the maximum flow is finite.

Give a bound on the running time.

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.
- Maximum matching: find a matching with as many edges as possible.



Reductions

- "Problem A reduces to problem B"
  - Rough meaning: there is a simple algorithm for A that uses an algorithm for B as a subroutine.
  - Denote $A \leq B$
- 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 identify problems where existing solutions may be applied.
  - Good programmers do this all the time

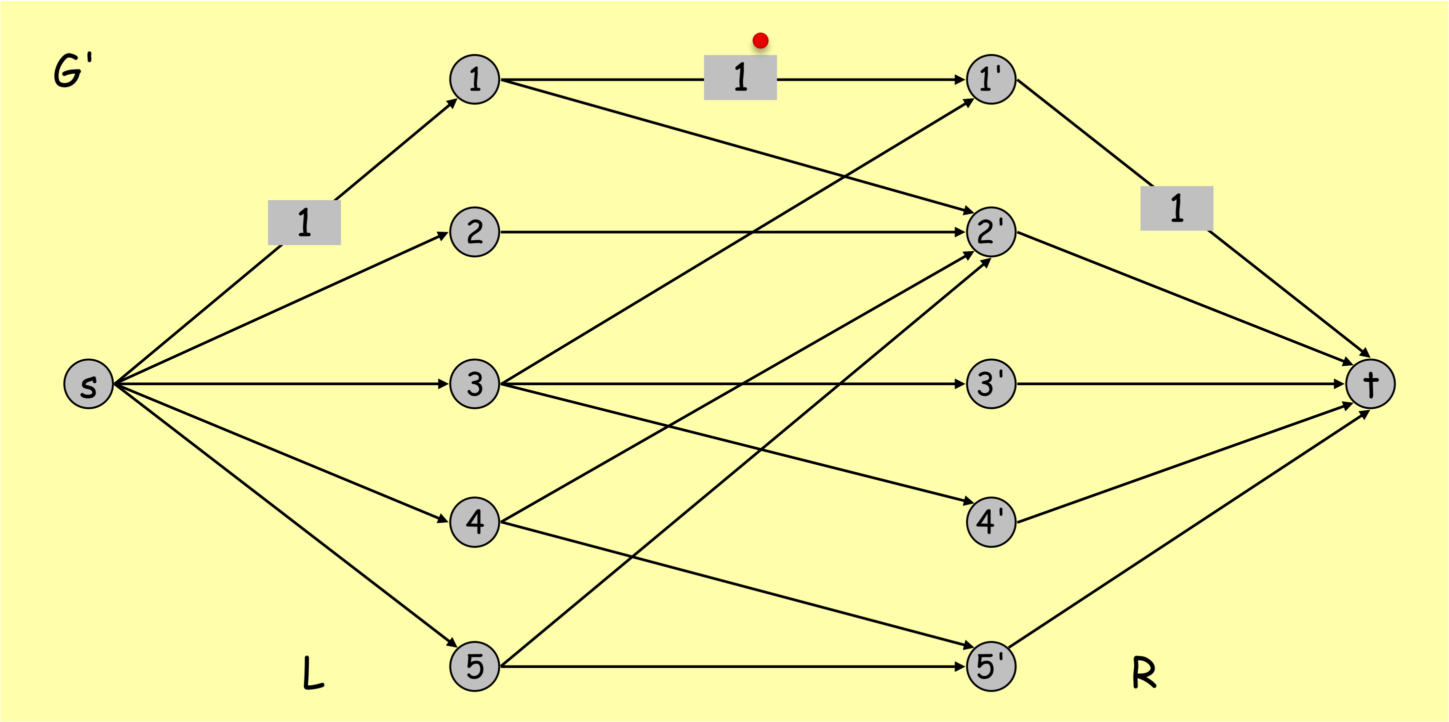
Reducing Bipartite Matching to Maximum Flow

Could also make capacities in the middle ∞

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.

Run FF and return the edges between $L \cup R$ carrying flow.



Bipartite Matching: Proof of Correctness

Theorem. # edges in max matching in G = value of max flow in G'.

Proof: We need two statements

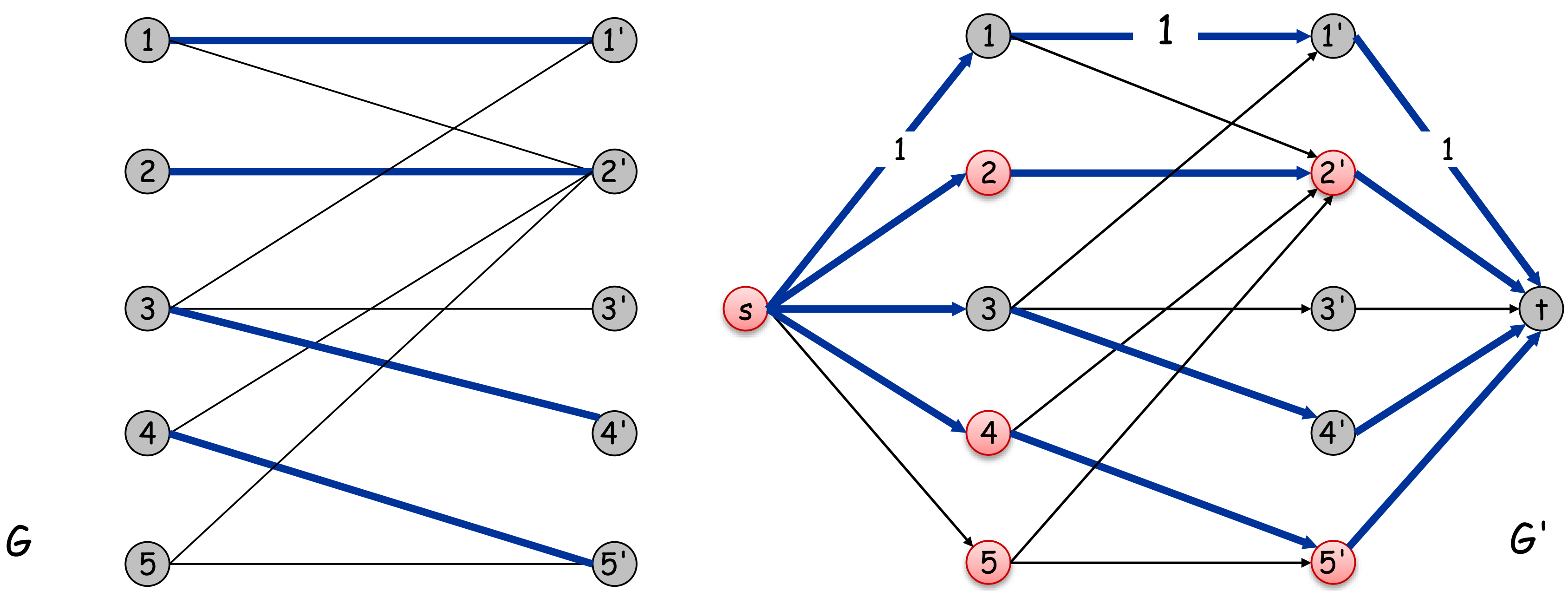
- # edges in max. matching in $G \leq$ max flow in G'
- # edges in max. matching in $G \geq$ max flow in G'

Bipartite Matching: Proof of Correctness

Theorem. # edges in max matching in G = value of max flow in G'.

Proof. $\leq$

- Given max matching M of cardinality k.
- Consider f sending 1 unit along path (s,u,v,t) for each (u,v)$\in M$.
- f is a flow, and has value k. ▪

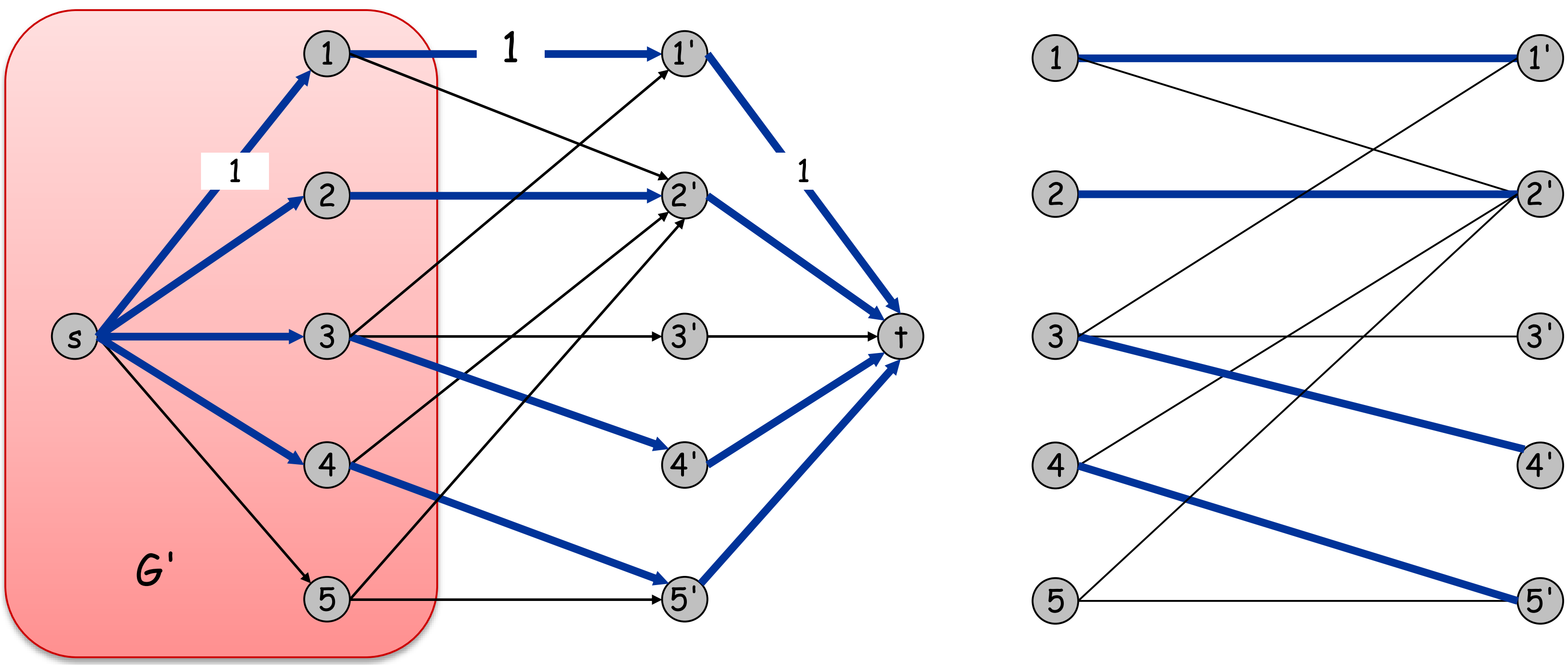


Bipartite Matching: Proof of Correctness

Theorem. # edges in max matching in G = value of max flow in G'.

Proof. $\geq$

- Let f be a max flow in G' of value k.
- Integrality theorem $\Rightarrow$ we can find a max flow f that is integral;
  - all capacities are 1 $\Rightarrow$ can find f that takes values only in {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
    - ✐ Because all capacities are 1 and flow must be conserved
  - |M| = k: consider flow across cut ($L \cup \{s\}$, $R \cup \{t\}$) ▪



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?

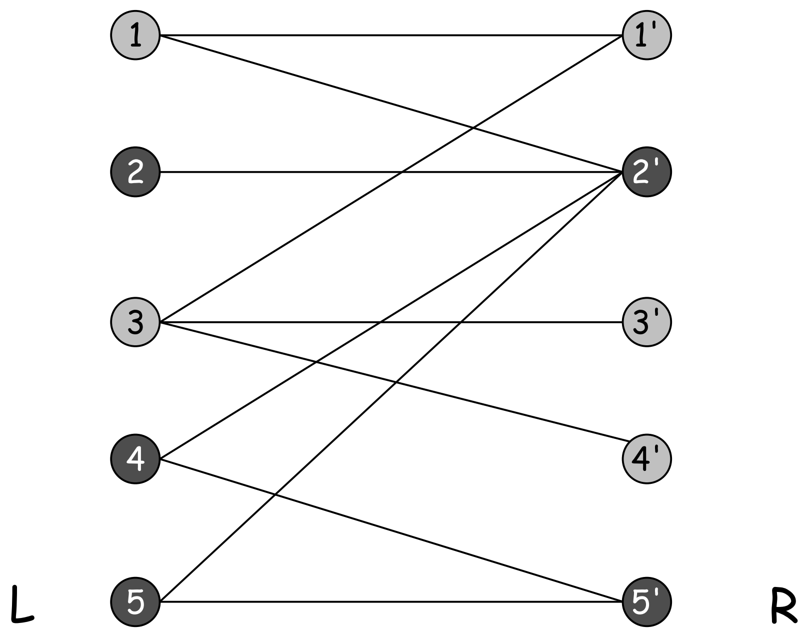
Structure of bipartite graphs with perfect matchings.

- 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 $S \subseteq L$.
Proof. Each node in S has to be matched to a different node in N(S).



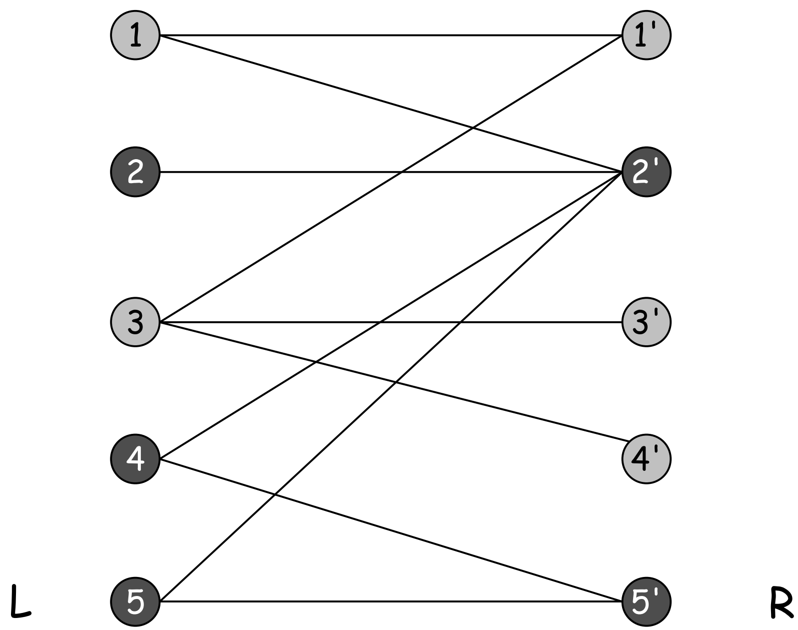
No perfect matching:
S = { 2, 4, 5 }
N(S) = { 2', 5' }.

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

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

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

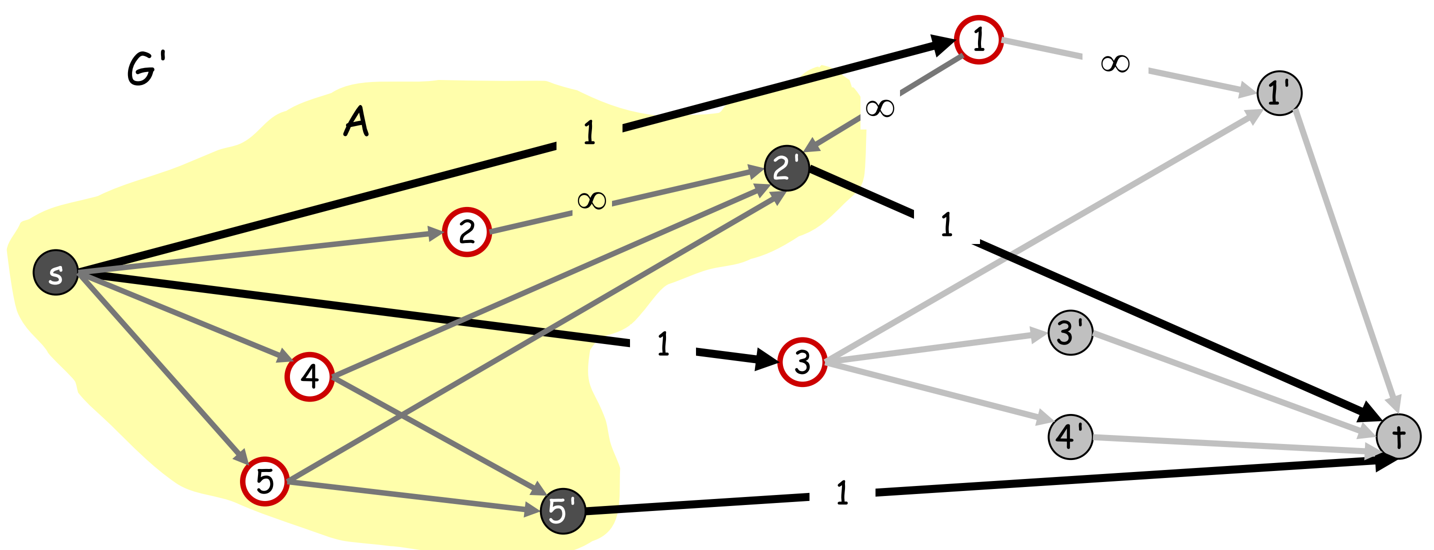


No perfect matching:
$S = \{ 2, 4, 5 \}$
$N(S) = \{ 2', 5' \}$.

Proof of Marriage Theorem

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

- Formulate as a max flow problem with ∞ capacities on edges from L to R and let (A, B) be min cut in G'.
- **Key property #1** of this graph: min-cut cannot use ∞ edges.
  - ✎ So $cap(A, B) = |L \cap B| + |R \cap A|$
- **Key property #2**: integral flow corresponds to a matching, as before.
  - By max-flow min-cut, cap(A, B) = (size of maximum matching) < | L |.
- Choose $S = L \cap A$.
  - 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}

$L \cap B = \{1, 3\}$

$R \cap A = \{2', 5'\}$

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

Bipartite Matching: Running Time

Which max flow algorithm to use for bipartite matching?

- Ford-Fulkerson: $O(mn)$.
- Capacity scaling: $O(m^2 \log C) = O(m^2)$.
- Shortest augmenting path (not covered in class): $O(m\, n^{1/2})$.
- Recent progress: $\tilde{O}(m^{10/7})$ [Madry, 2013]

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]
- Better algorithms for dense graphs, e.g. $O(n^{2.38})$ [Harvey, 2006]

Review Question

- A bipartite graph is k-regular if |L|=|R| and every vertex (regardless of which side it is on) has exactly k neighbors

- Prove or disprove: every k-regular bipartite graph has a perfect matching

7.6 Disjoint Paths

Application of Max Flow With C=1

Two problems

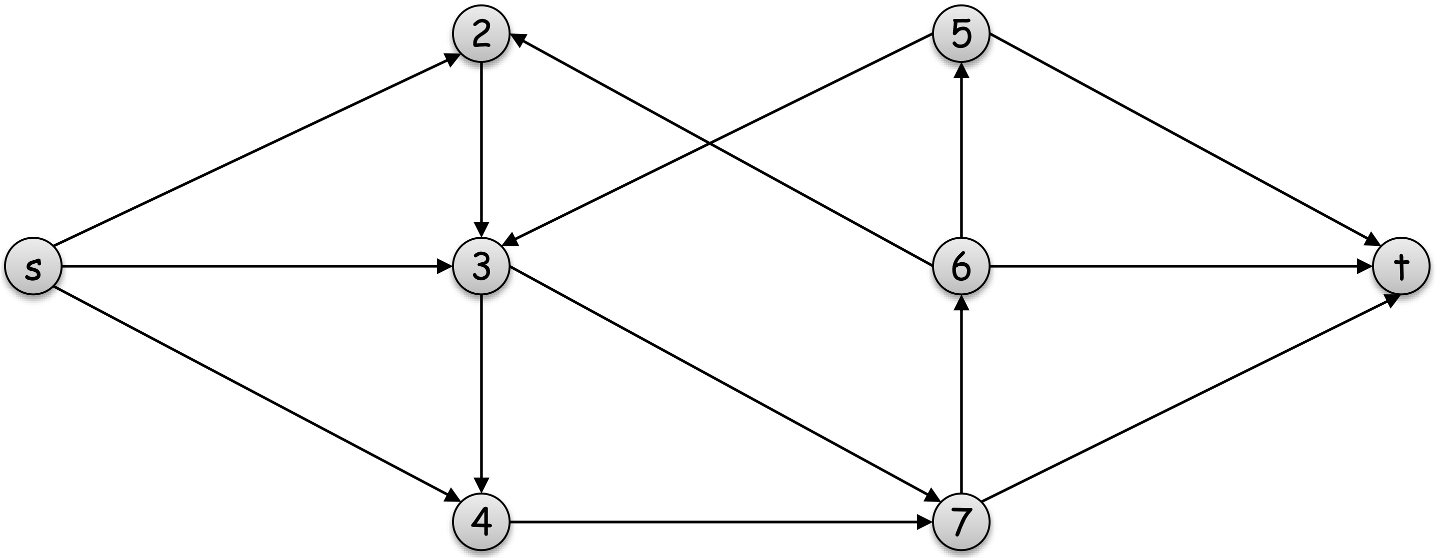
Given a network:

- Find edge-disjoint paths

- Find how many edges must be deleted to disconnect the graph

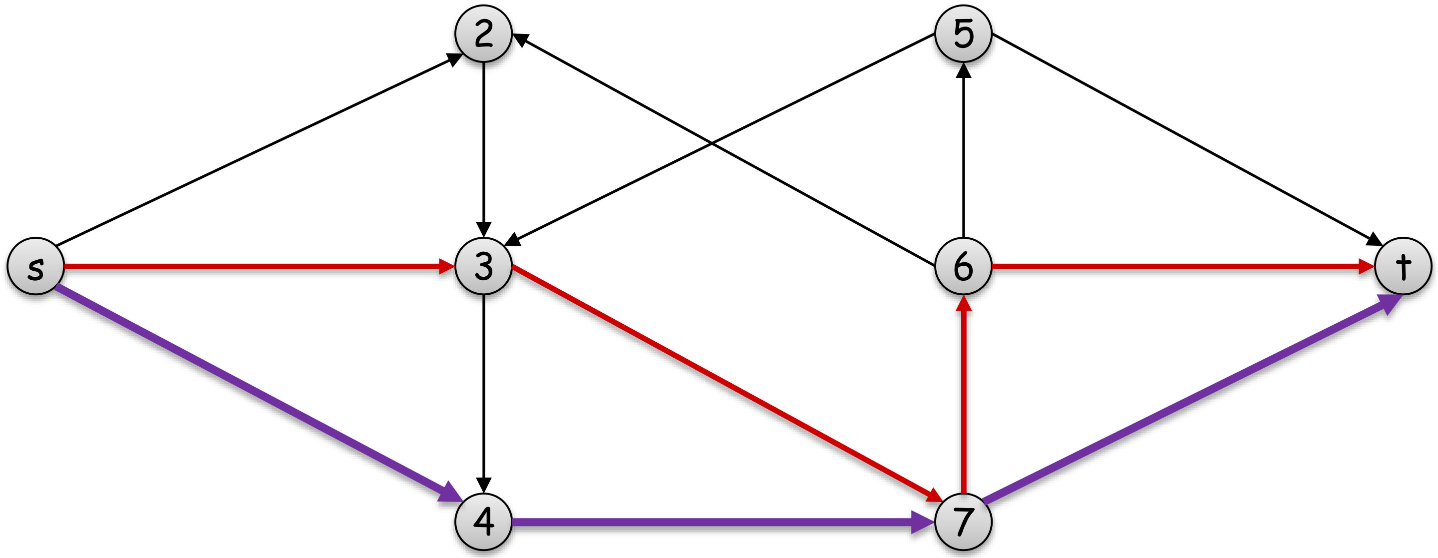
Edge Disjoint Paths

- **Disjoint paths problem.** Given a digraph $G = (V, E)$ and two nodes s and t, find the max number of edge-disjoint s-t paths.
  - Two paths are edge-disjoint if they have no edge in common.
  - In networks: how many packets can I send in parallel?



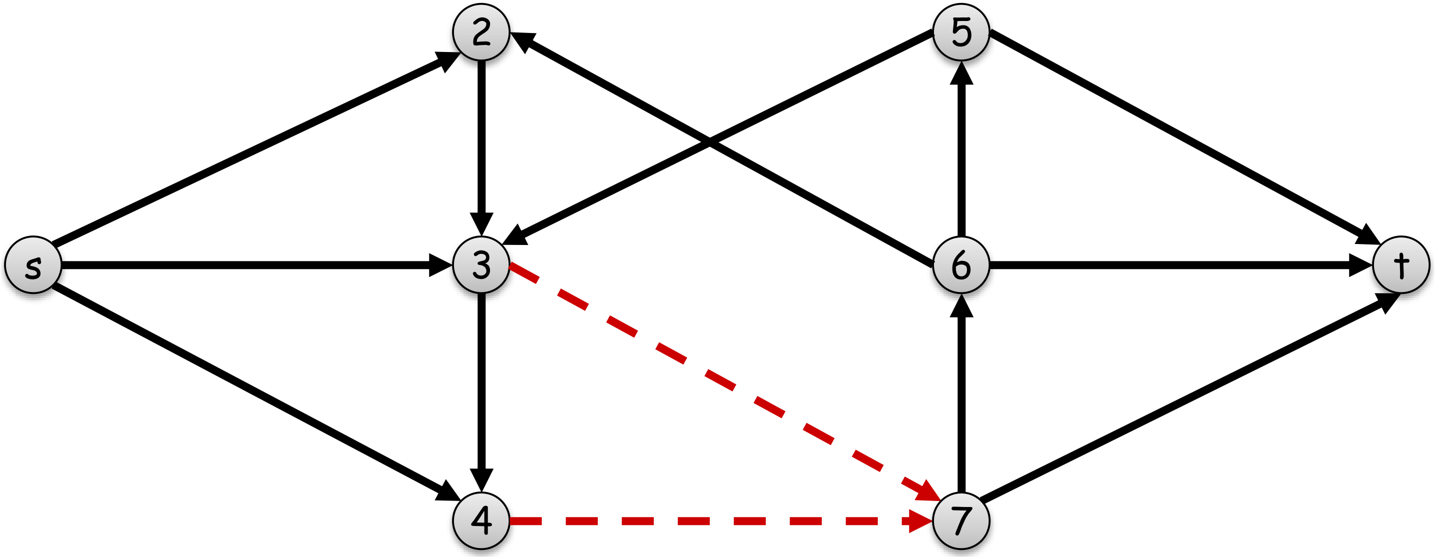
Edge Disjoint Paths

- **Disjoint paths problem.** Given a digraph $G = (V, E)$ and two nodes s and t, find the max number of edge-disjoint s-t paths.
  - Two paths are edge-disjoint if they have no edge in common.
  - In networks: how many packets can I send in parallel?



Network Connectivity

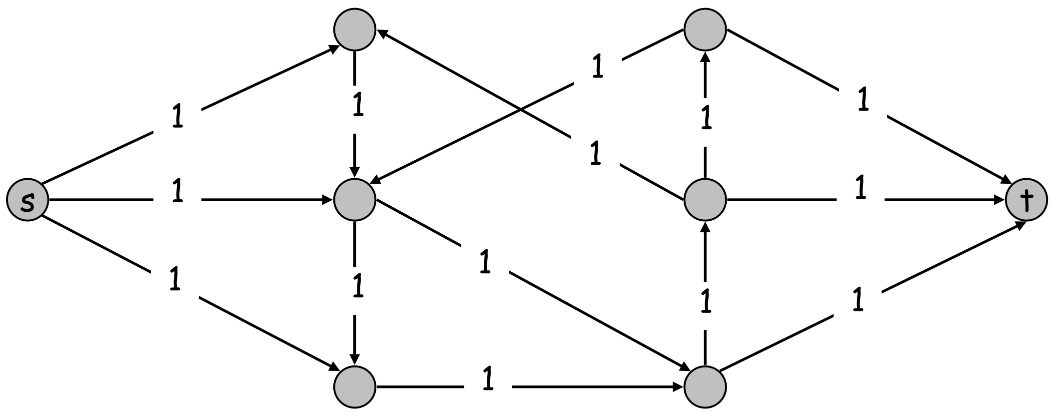
- **Network connectivity problem.** Given a digraph G = (V, E) and two nodes s and t, find min number of edges whose removal disconnects t from s.
  - A set of edges $F \subseteq E$ disconnects t from s if each s-t paths uses at least one edge in F.
    (That is, removing F would make t unreachable from s.)



How is it related to edge-disjoint paths?

Edge Disjoint Paths

Max flow formulation: assign unit capacity to every edge.



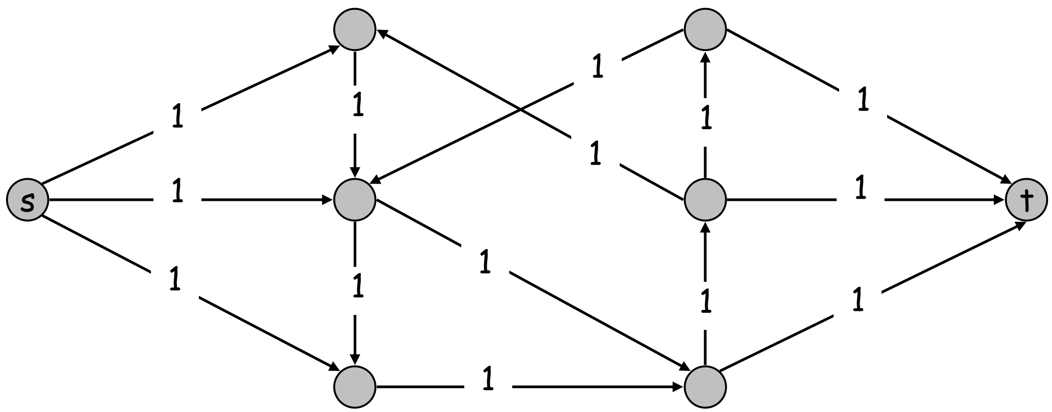
Theorem. Max number edge-disjoint s-t paths equals max flow value.

Proof. $\leq$

- Suppose there are k edge-disjoint paths $P_1, \ldots, P_k$.
- Set $f(e) = 1$ if e participates in some path P_i ; else set $f(e) = 0$.
- 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. Max number edge-disjoint s-t paths equals max flow value.

Proof. $\geq$

- Suppose max flow value is k.
- Integrality theorem $\Rightarrow$ there exists 0-1 flow f of value k.
- Consider edge (s, u) with f(s, u) = 1.
  - by 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. ▪

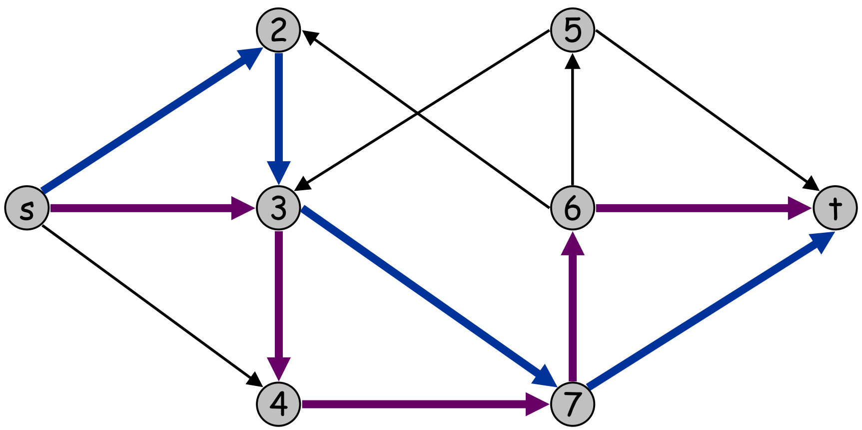
can eliminate cycles to get simple paths if desired

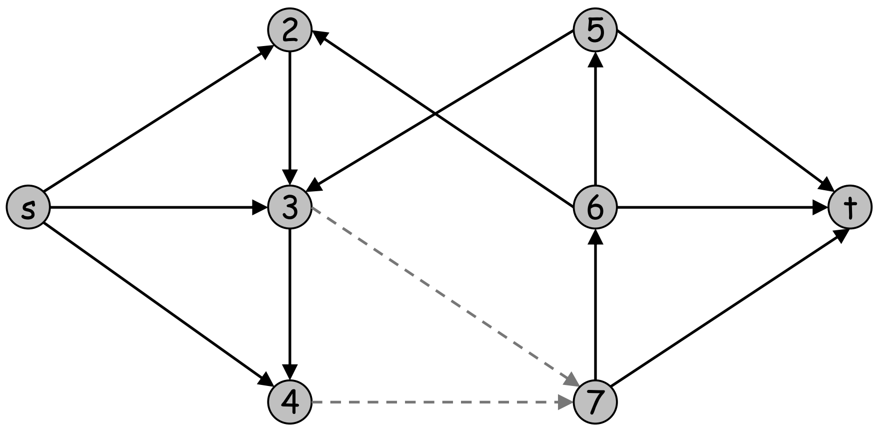
Edge Disjoint Paths and Network Connectivity

Theorem. [Menger 1927] The max number of edge-disjoint s-t paths is **equal** to the min number of edges whose removal disconnects t from s.

Proof. $\leq$

- Suppose the removal of $F \subseteq E$ disconnects t from s, and $|F| = k$.
- All s-t paths use at least one edge of F. Hence, the number of edge-disjoint paths is at most k. ▪





Disjoint Paths and Network Connectivity

Theorem. [Menger 1927] The max number of edge-disjoint s-t paths is equal to the min number of edges whose removal disconnects t from s.

Pf. $\geq$

- Suppose max number of edge-disjoint paths is k.
- Then max flow value is k.
- Max-flow min-cut $\Rightarrow$ cut (A, B) of capacity k.
- Let F be set of edges going from A to B.
- |F| = k and disconnects t from s. ▪