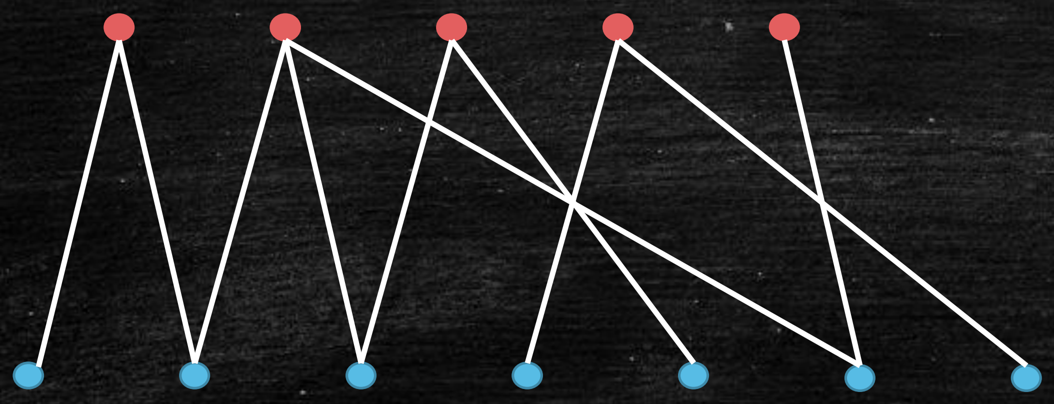
Applications of Max-Flow

Matching (in Bipartite Graphs)

Maximum Bipartite Matching

- Top vertices are girls, bottom vertices are boys.
- An edge represent a possible match for a boy and a girl.
- Problem: find a maximum matching for boys and girls.

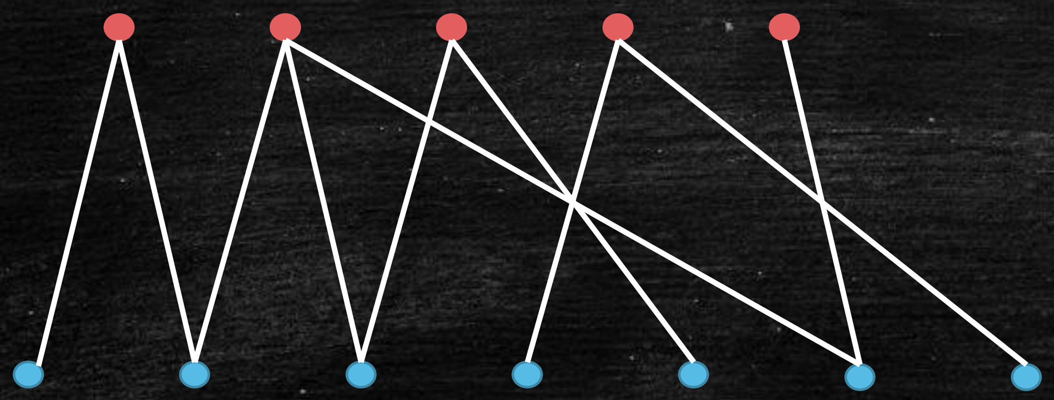


Maximum Bipartite Matching - Formal

- Given a graph $G = (V, E)$, a matching M is a subset of edges that do not share vertices in common.
- The size of a matching is the number of edges in it.
- Problem: Given a bipartite graph $G = (A, B, E)$ find a matching with the maximum size.

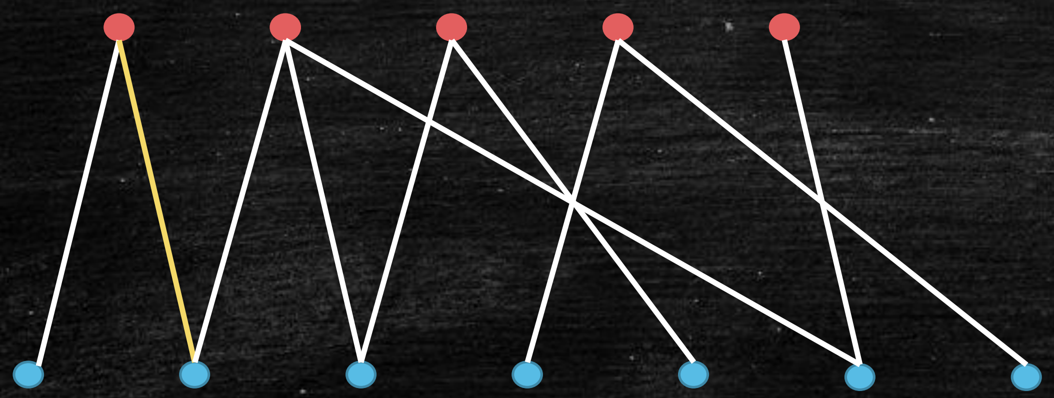
Maximum Bipartite Matching

- Naïve greedy doesn't work!



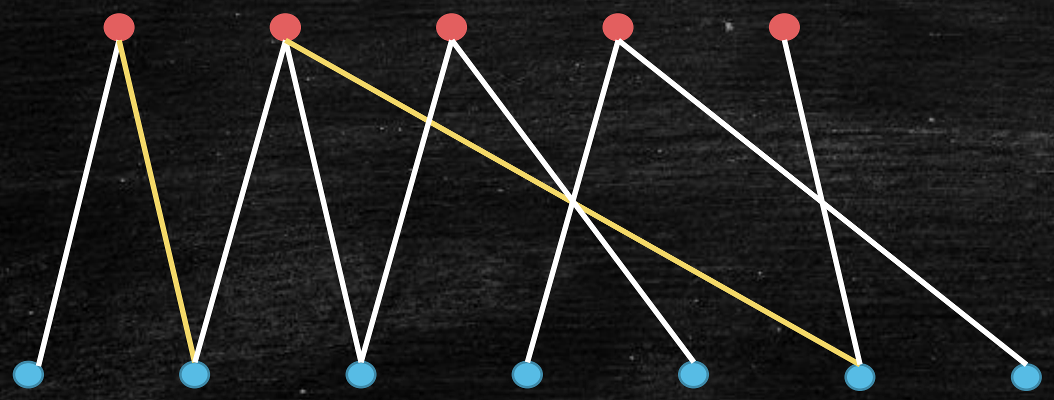
Maximum Bipartite Matching

- Naïve greedy doesn't work!



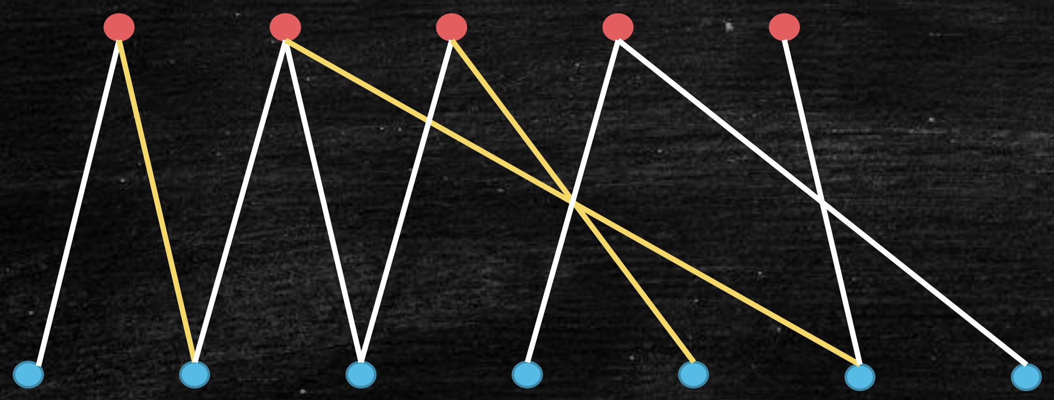
Maximum Bipartite Matching

- Naïve greedy doesn't work!



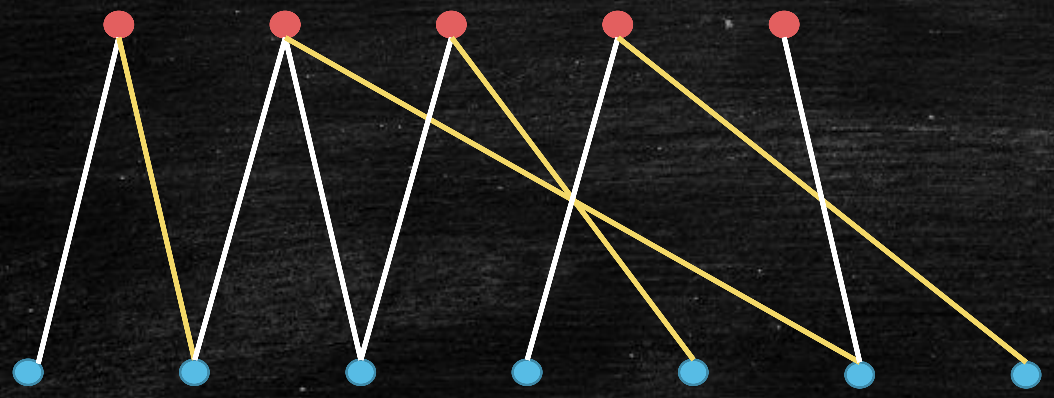
Maximum Bipartite Matching

- Naïve greedy doesn’t work!



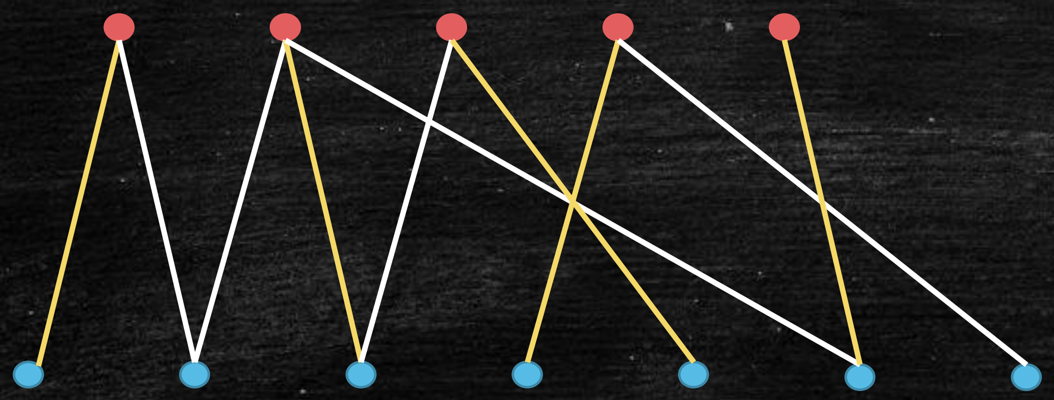
Maximum Bipartite Matching

- Naïve greedy doesn't work!
- A total of 4 matches...



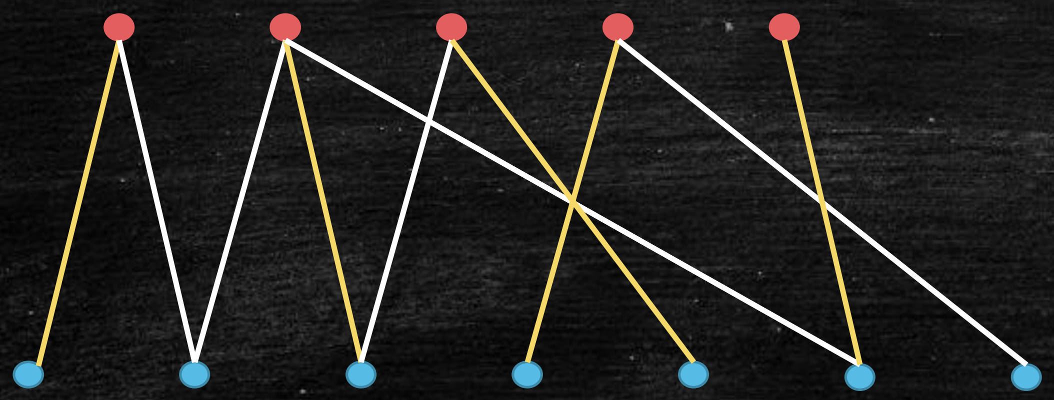
Maximum Bipartite Matching

- Naïve greedy doesn't work!
- A better solution…



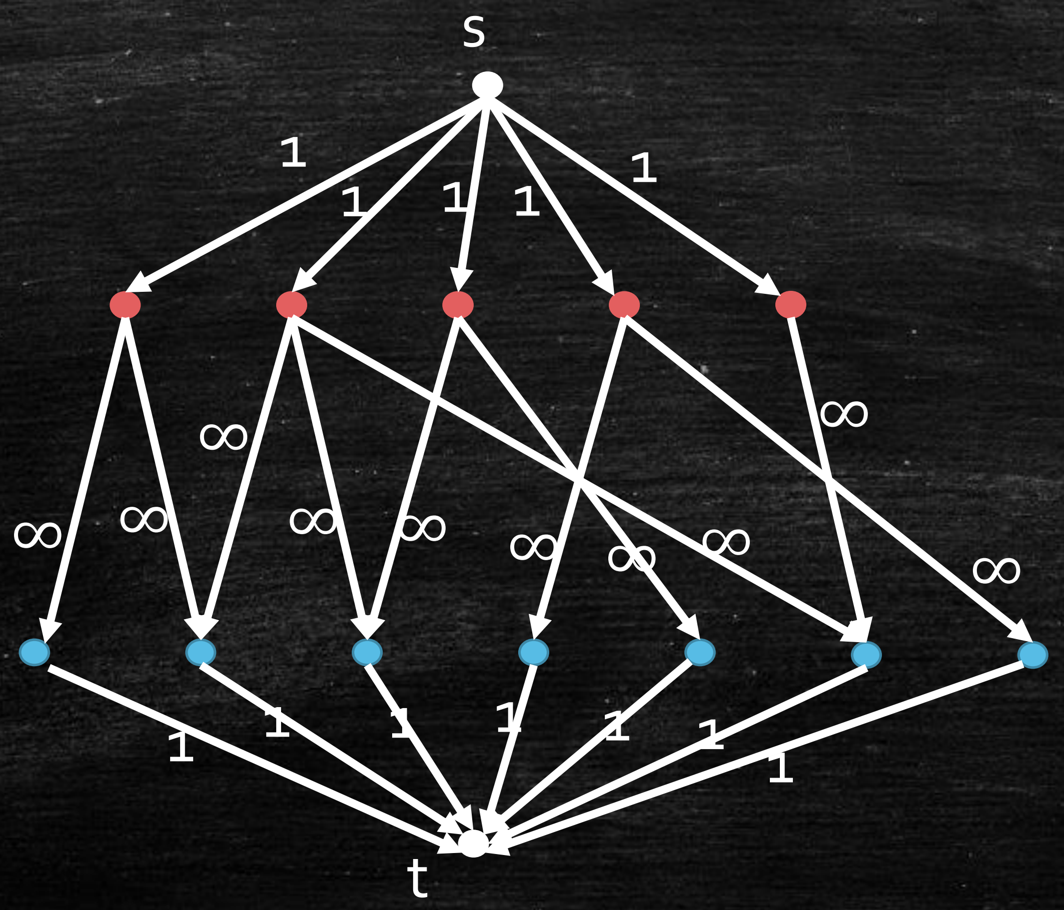
Maximum Bipartite Matching

- Naïve greedy doesn't work!
- A better solution…
- Greedy finds a maximal matching, not a maximum one!



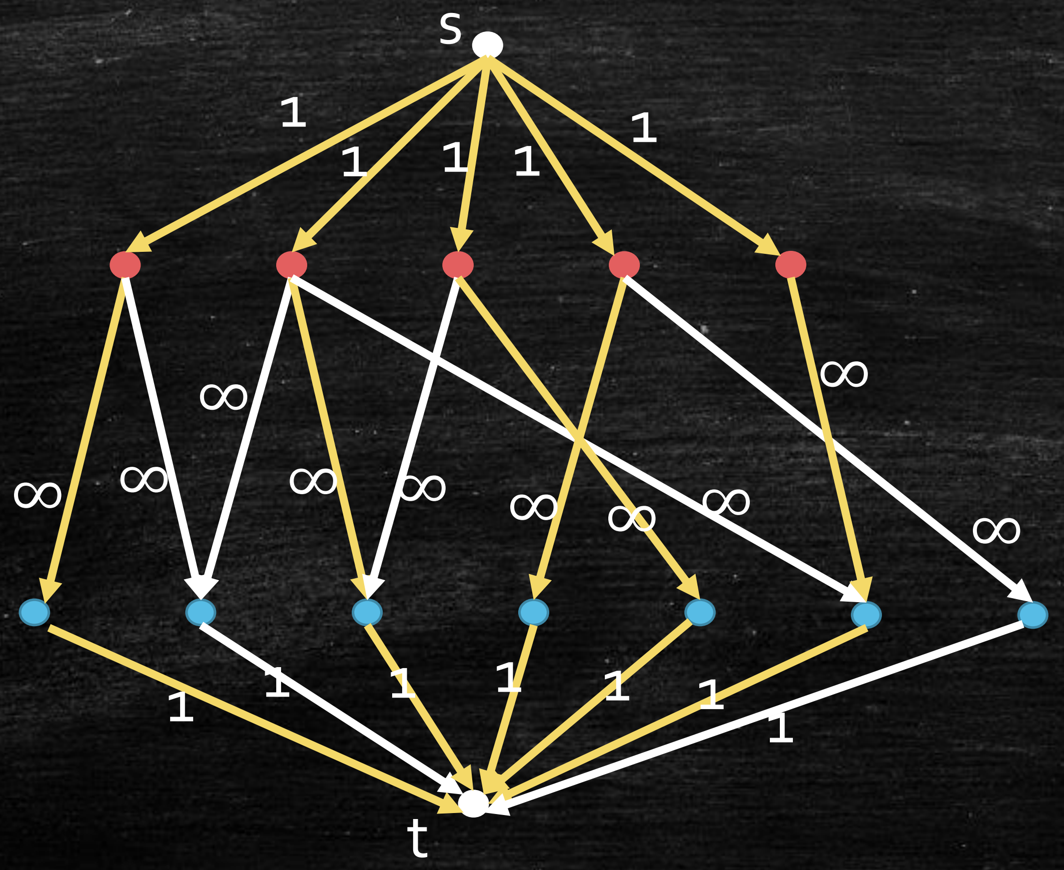
Maximum Bipartite Matching

- Applying maximum flow and Ford-Fulkerson Method.



Maximum Bipartite Matching

- An integral flow corresponds to a matching.
- Integrality theorem ensures the maximum flow can be integral.



Class Activity

- A graph is regular if all the vertices have the same degree.
- A matching is perfect if all the vertices are matched.

Let $G = (A, B, E)$ be a regular bipartite graph. Which of the followings is correct?

A. We always have $|A| = |B|$, but G may not contain a perfect matching

B. We always have $|A| = |B|$, and G always contains a perfect matching

C. It is possible that $|A| < |B|$, but G always contains a matching of size $|A|$

D. It is possible that $|A| < |B|$, and the maximum matching in G may have size less than $|A|$.

Matching (General)

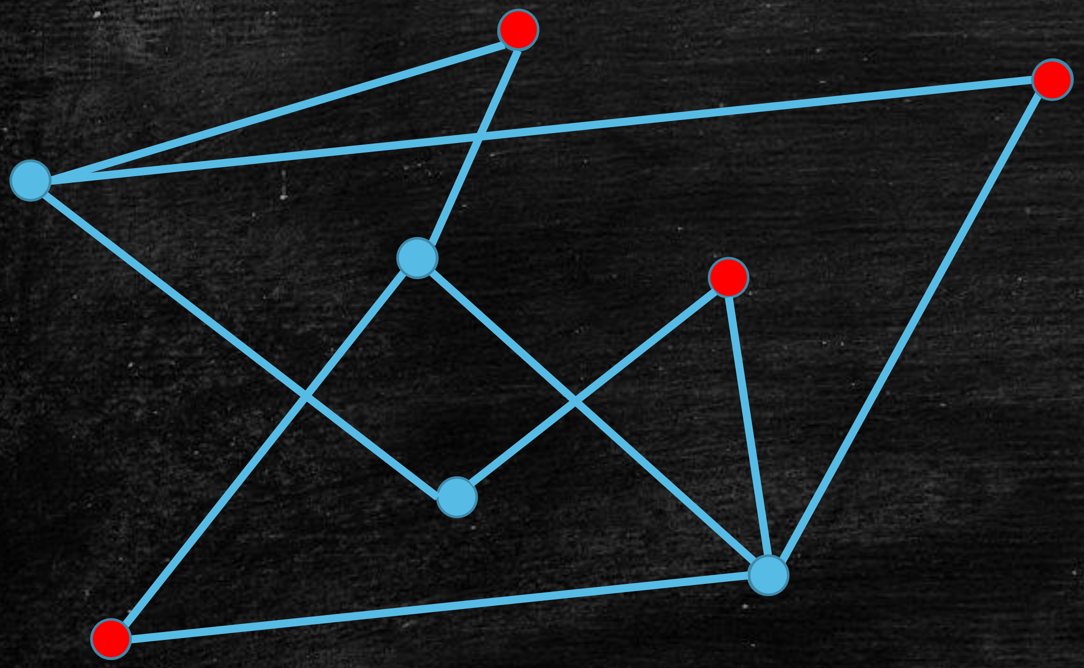
- Maximum Matching in general graphs?
- Edmonds' Blossom Algorithm, $O(|E| \cdot |V|^2)$
- Maximum Weighted Matching in bipartite graphs?
- Hungarian Algorithm, $O(|V|^3)$
- Maximum Weighted Matching in general graphs?
- A clever algorithm that combines Edmonds' Blossom Algorithm and Hungarian Algorithm, $O(|V|^3)$

Max-Flow-Min-Cut Revisited

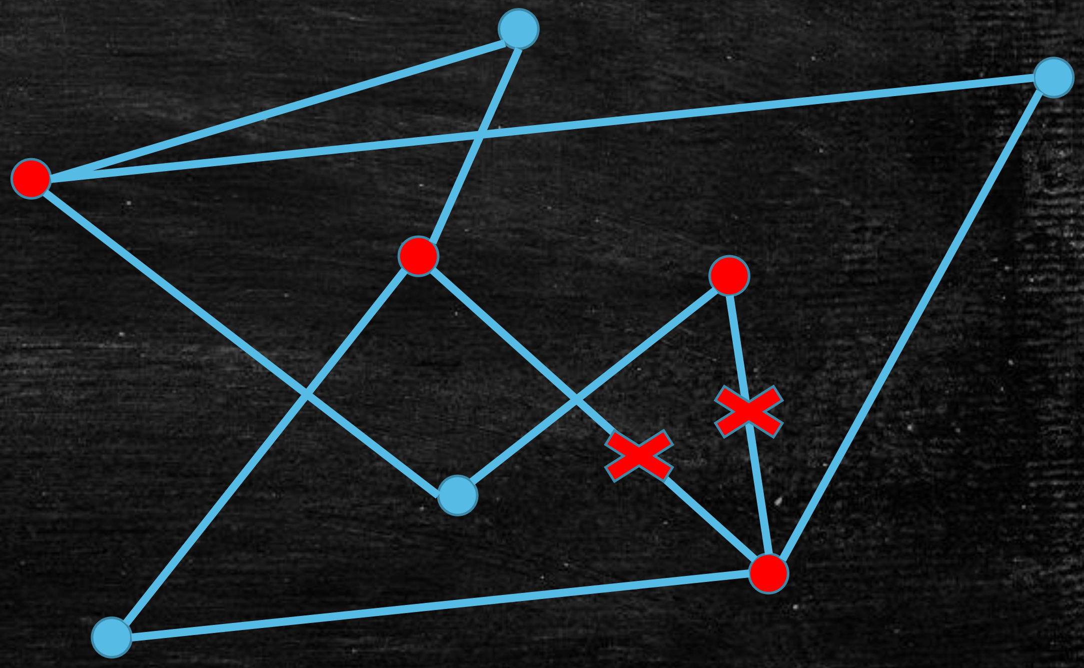
Independent Set and Vertex Cover on Bipartite Graphs

Independent Set

- Given an undirected graph $G = (V, E)$, a subset of vertices $S \subseteq V$ is an independent set if there is no edge between any two vertices in S.



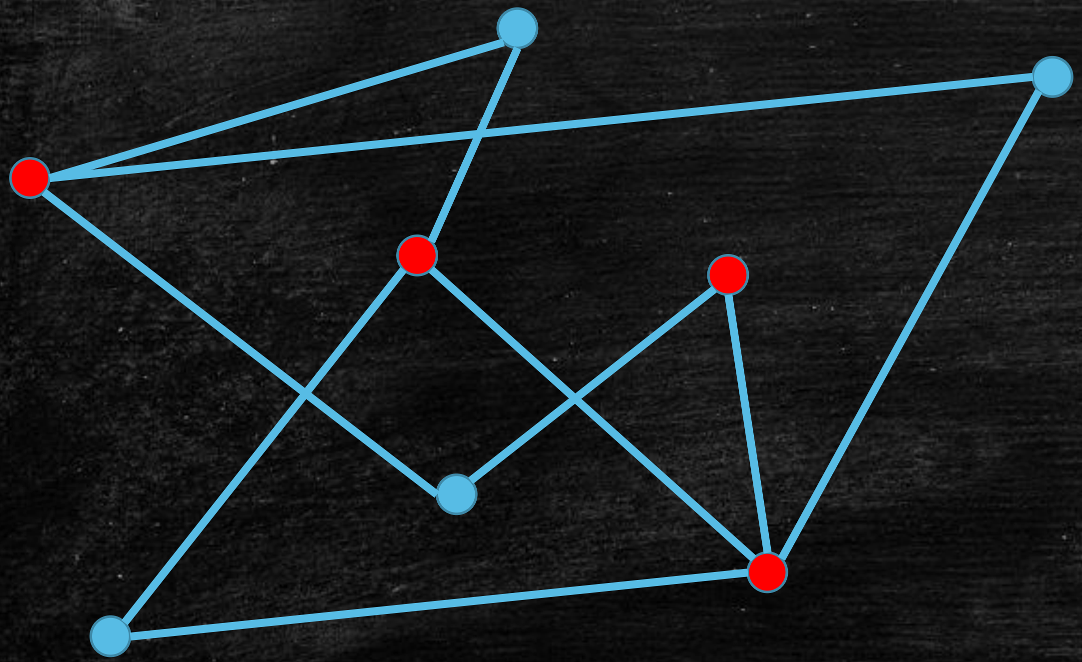
an independent set



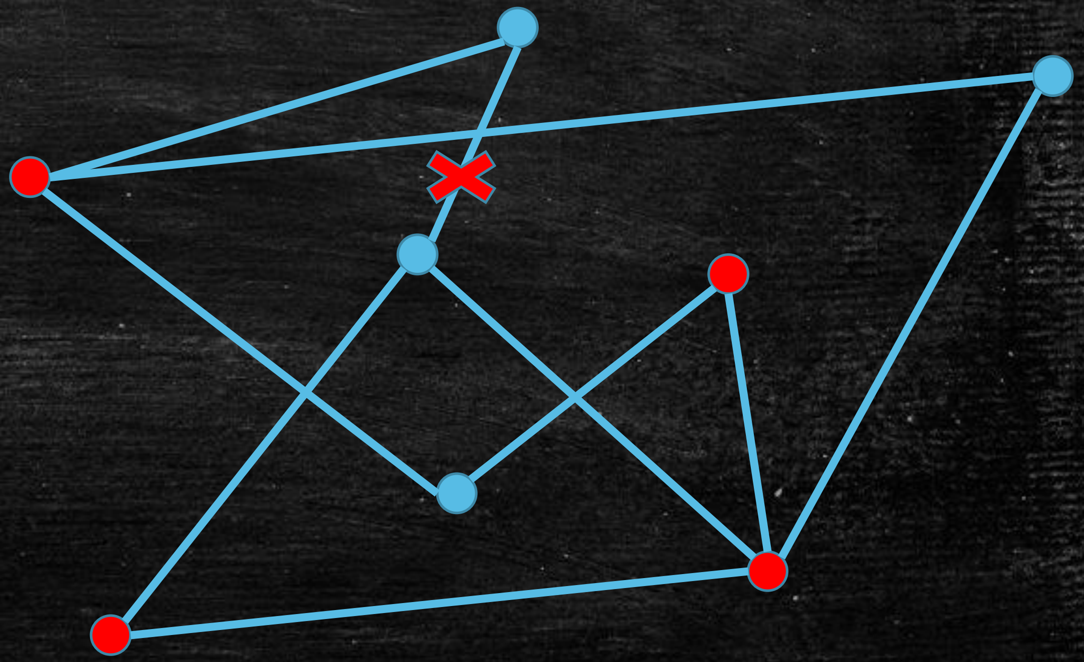
not an independent set

Vertex Cover

- Given an undirected graph $G = (V, E)$, a subset of vertices $S \subseteq V$ is a vertex cover if S contains at least one endpoint of every edge.



a vertex cover



not a vertex cover

Optimization

- [Maximum Independent Set] Given an undirected graph $G = (V, E)$, find an independent set with the maximum size.
- [Minimum Vertex Cover] Given an undirected graph $G = (V, E)$, find a vertex cover with the minimum size.

Exercise

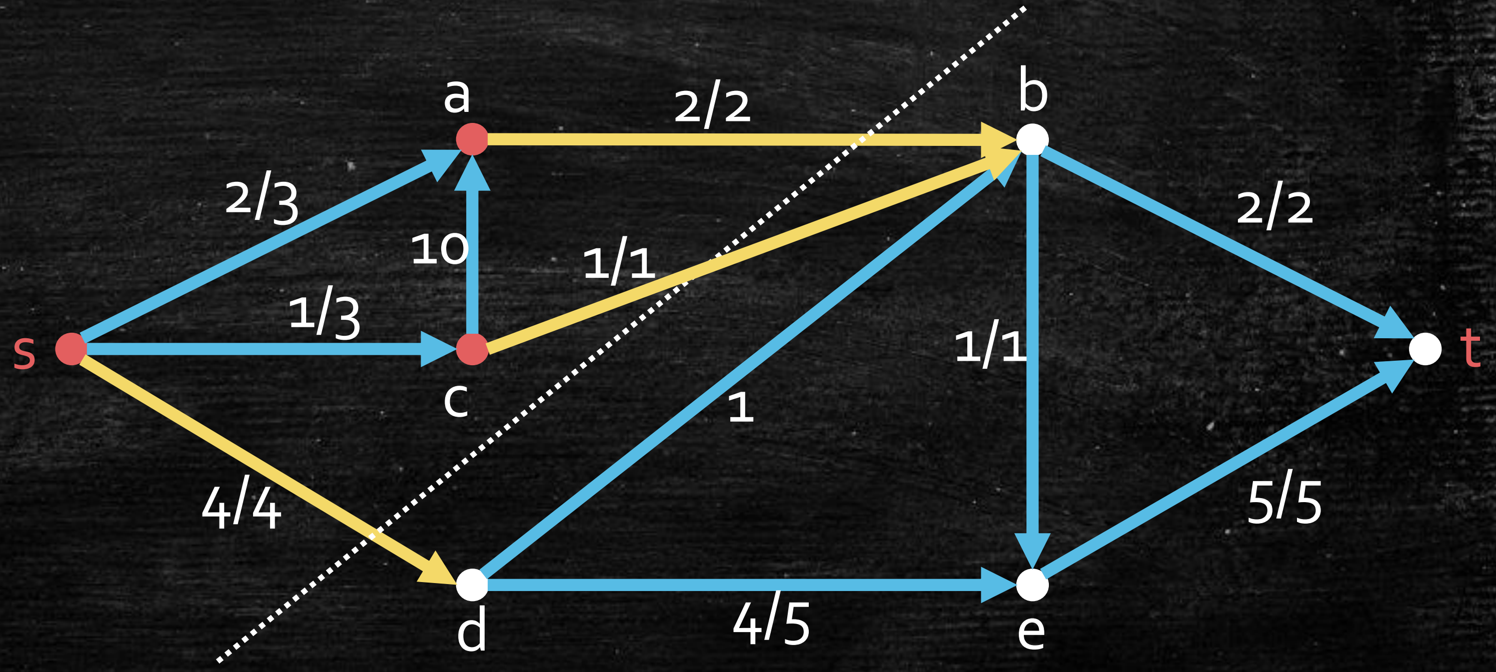
- Given an undirected graph $G = (V, E)$, prove that S is an independent set if and only if $V \setminus S$ is a vertex cover.

On Bipartite Graphs

- Both maximum independent set and minimum vertex cover are NP-hard!
- However, they are “easy” on bipartite graphs.
- Minimum Cut

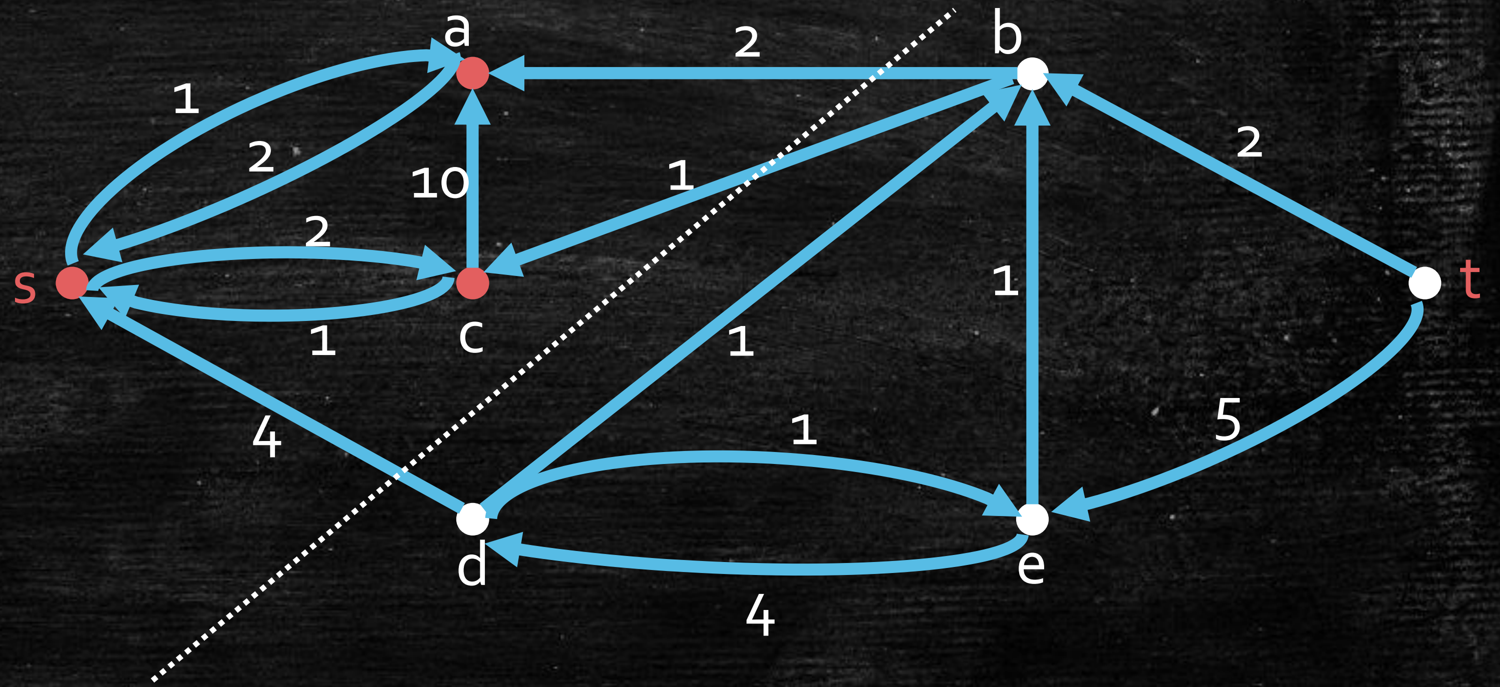
Max-Flow-Min-Cut Theorem Revisited

- Min-cut describes the “bottleneck” of max-flow
- Max-Flow = Min-Cut



Max-Flow-Min-Cut Theorem Revisited

- Min-cut describes the “bottleneck” of max-flow
- Max-Flow = Min-Cut
- No edge goes from s-side to t-side in residual network

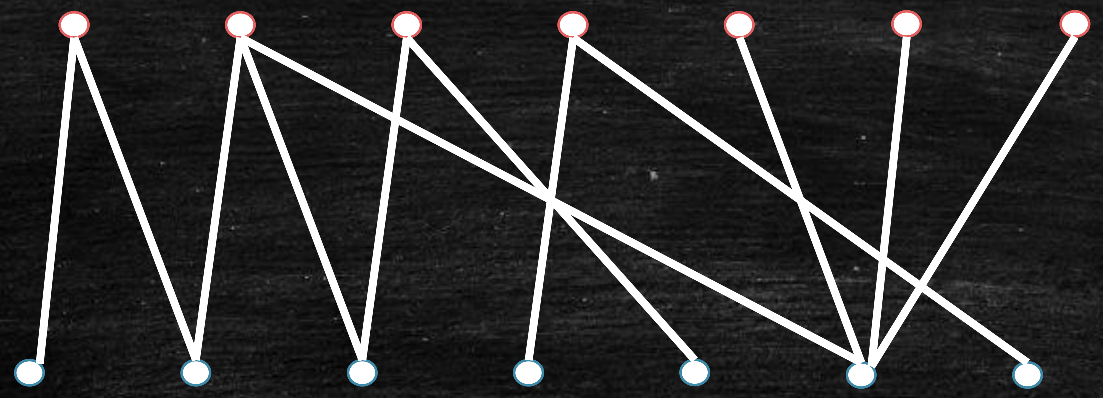


Max-Flow-Min-Cut Theorem Revisited

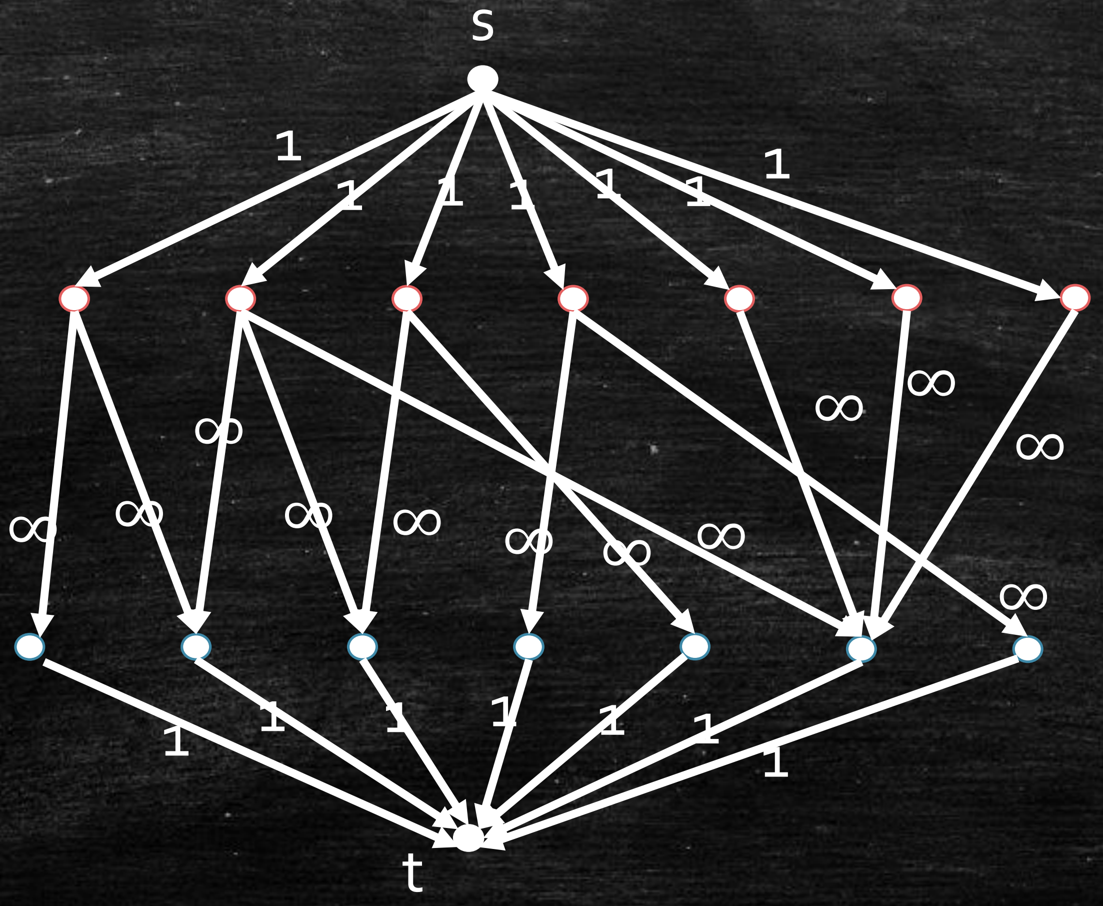
- Min-cut describes the “bottleneck” of max-flow
- Max-Flow = Min-Cut
- No edge goes from s-side to t-side in residual network
- Given a max-flow, the set of vertices reachable from s gives a min-cut

Let's look at this graph

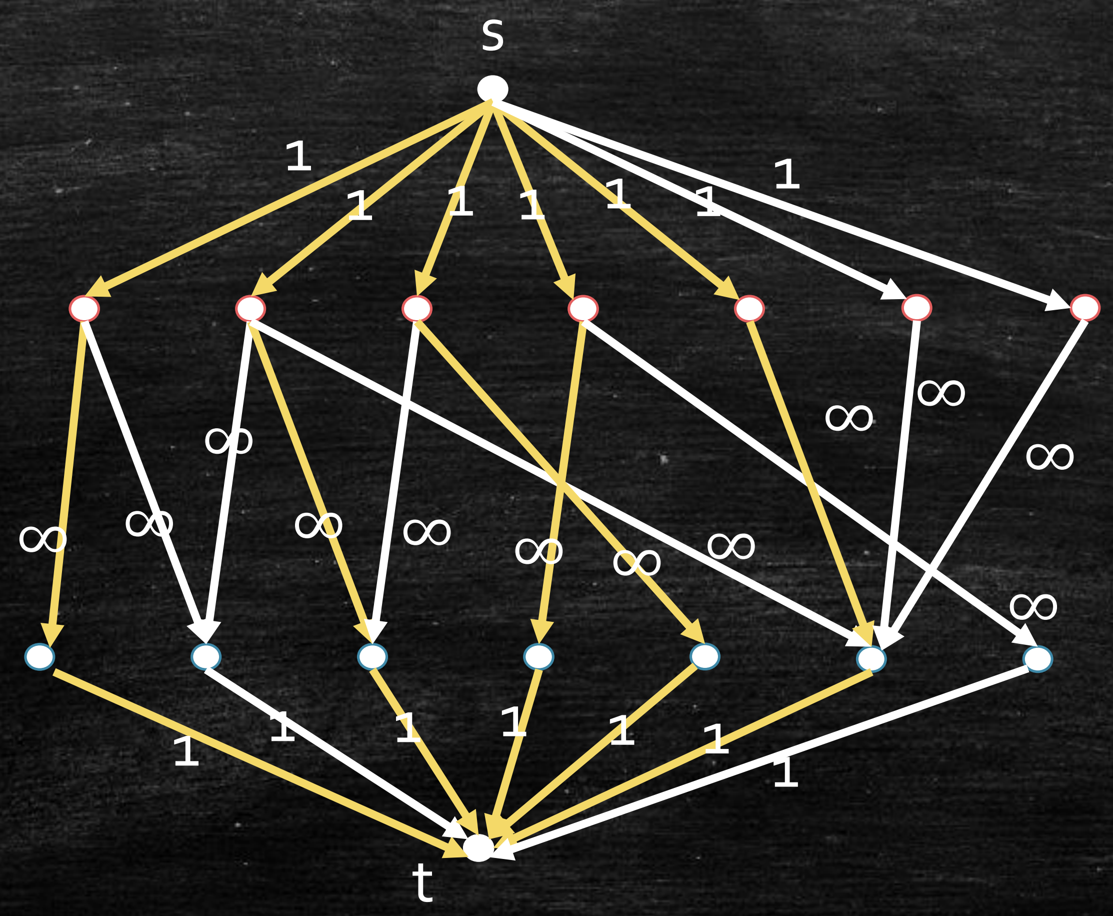
- Maximum Independent Set?
- Minimum Vertex Cover?



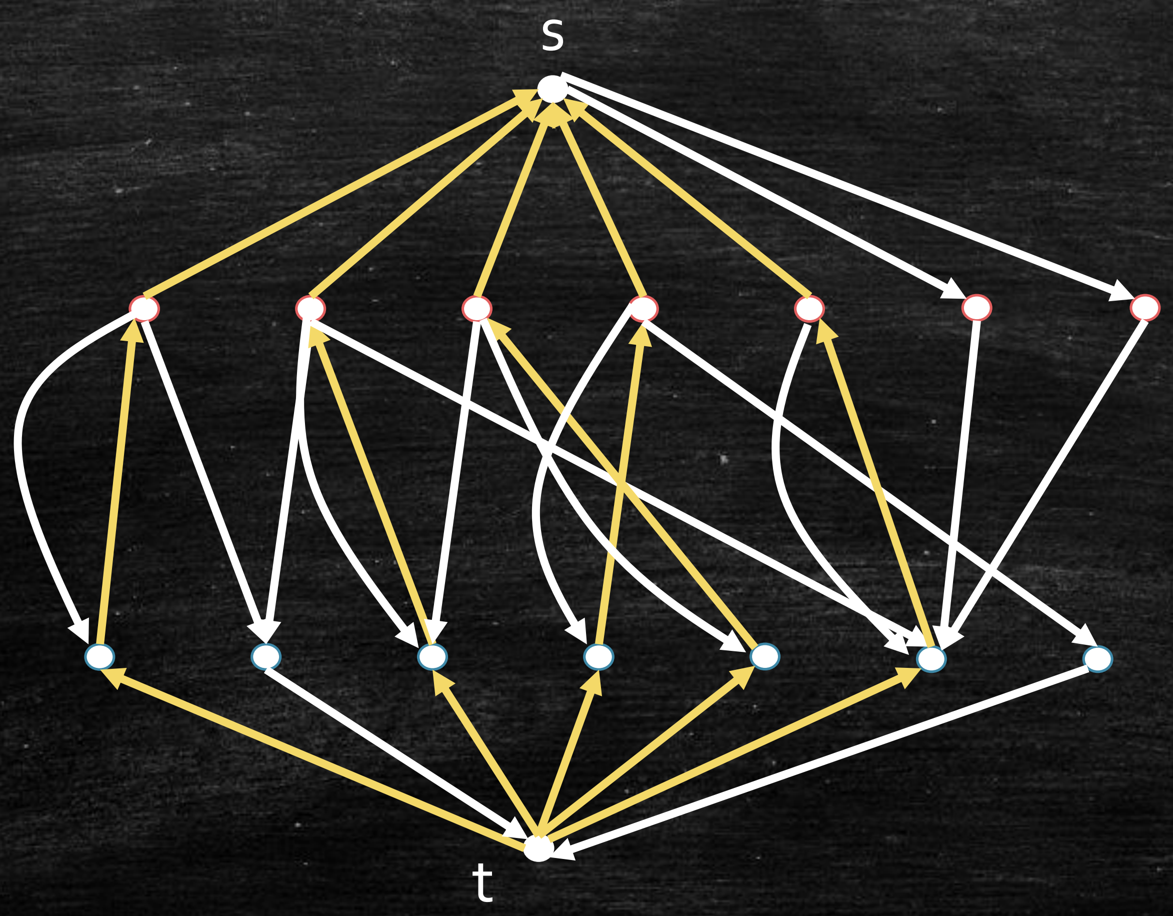
Convert it to a max-flow problem...



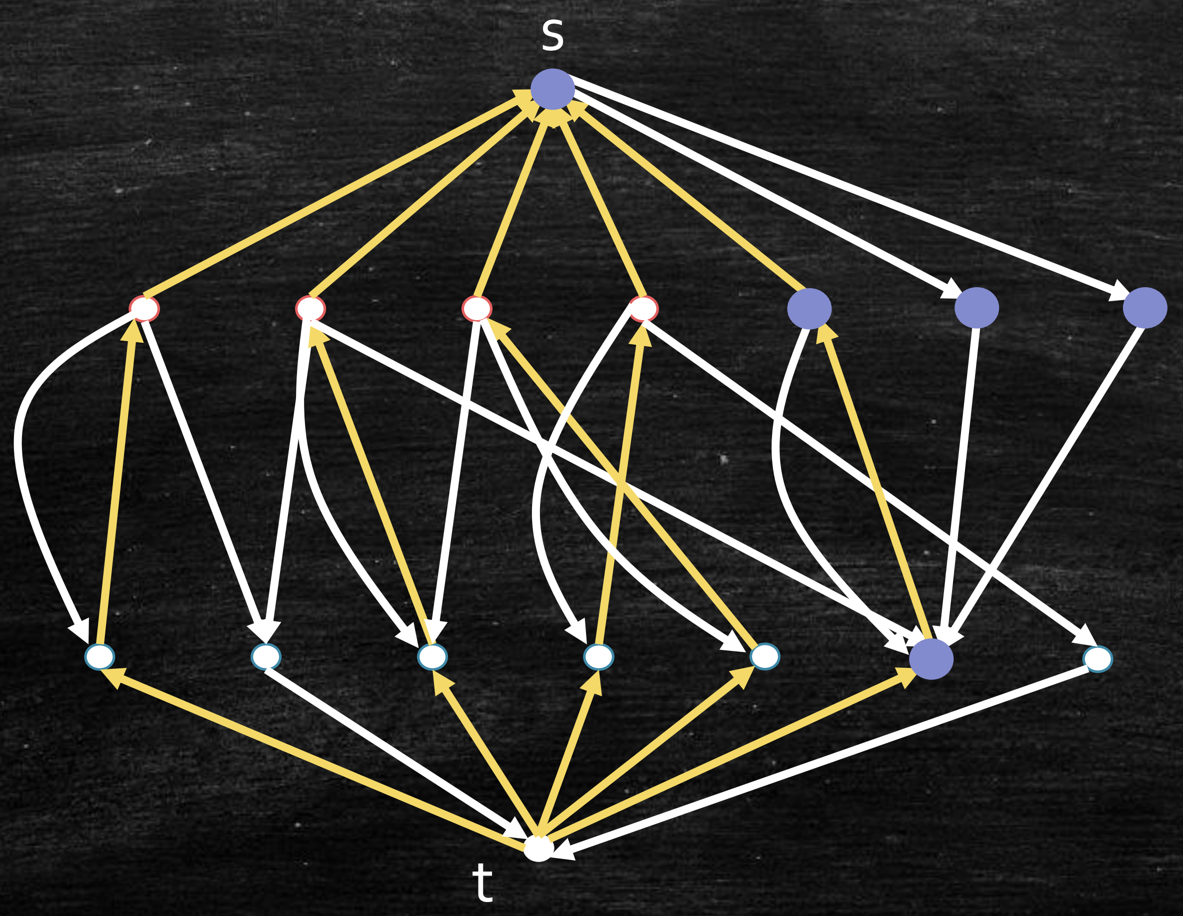
Max-Flow = 5



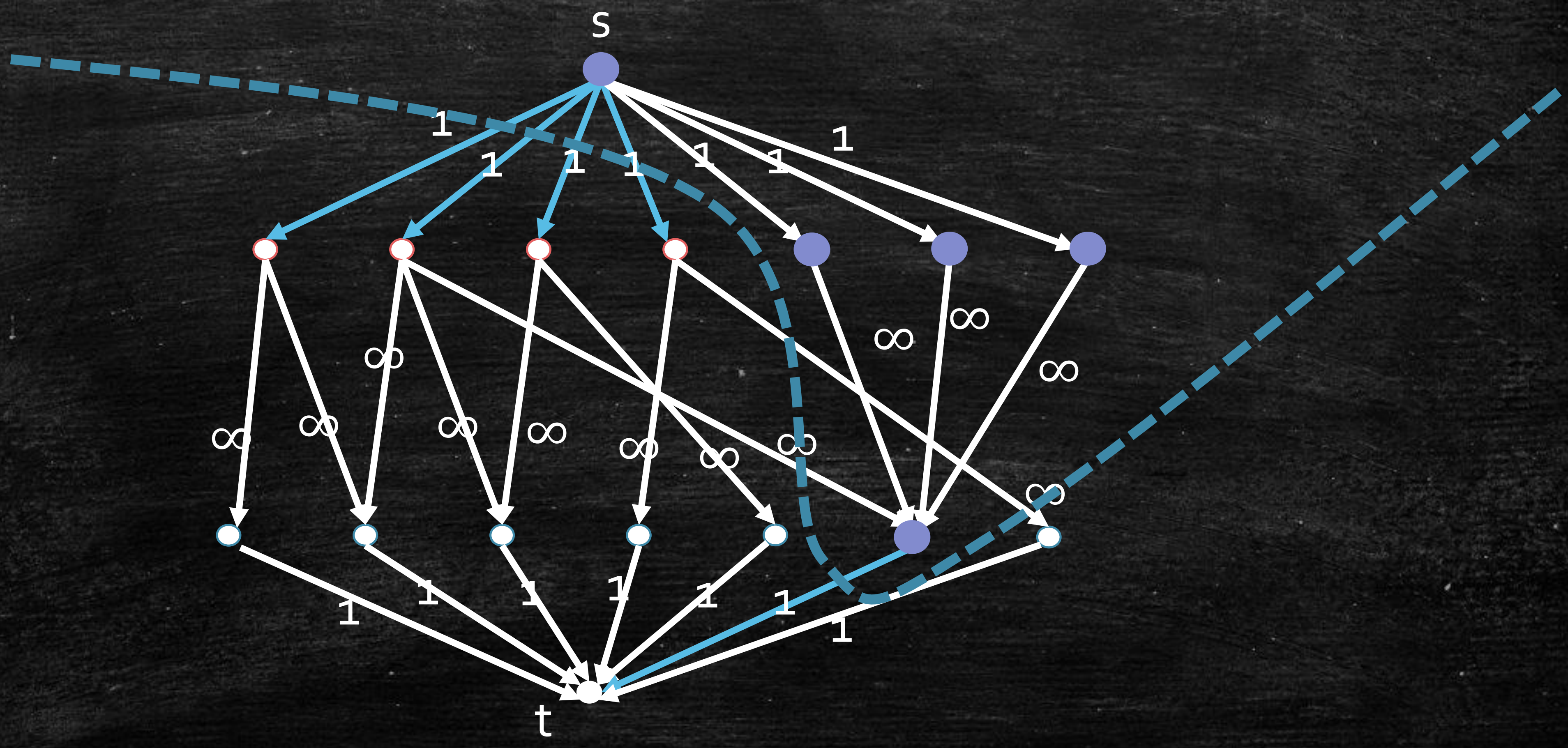
Residual Graph G^f



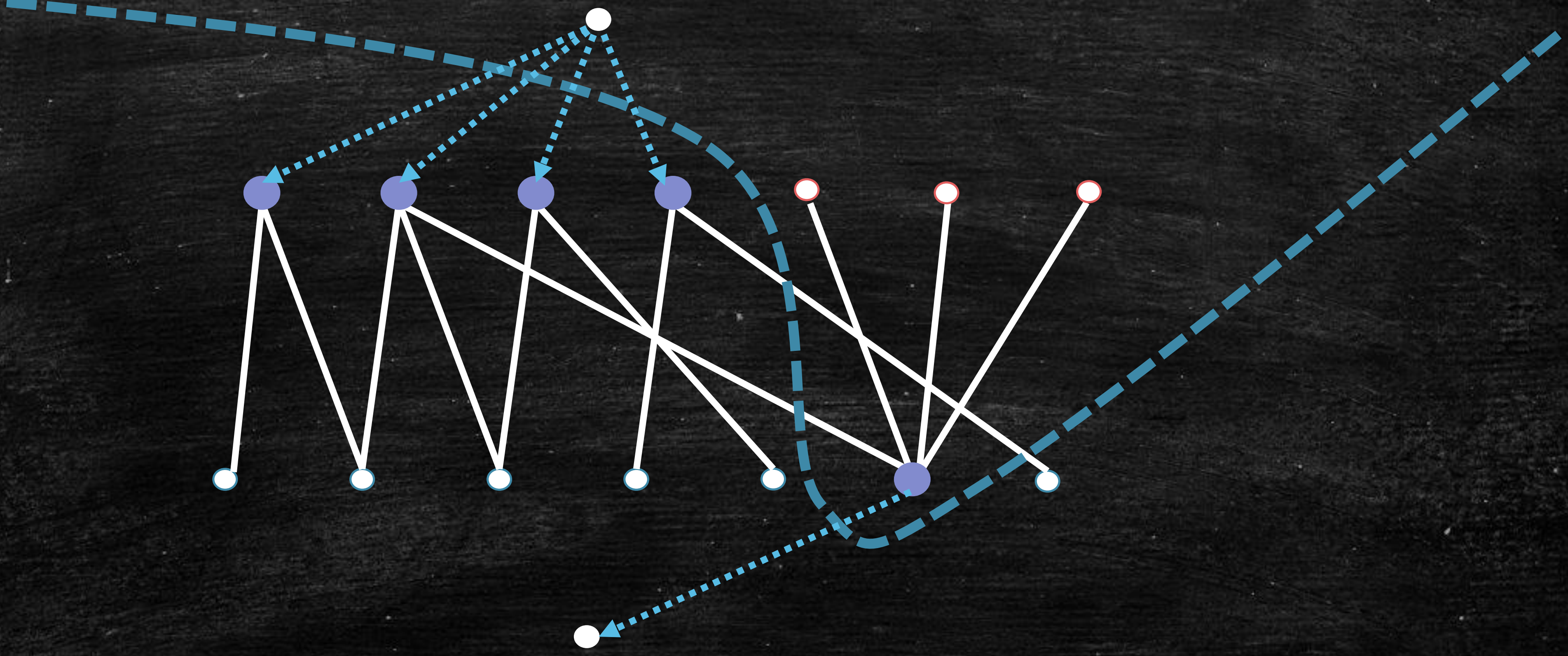
Vertices Reachable from s in G^f



Min-Cut = 5

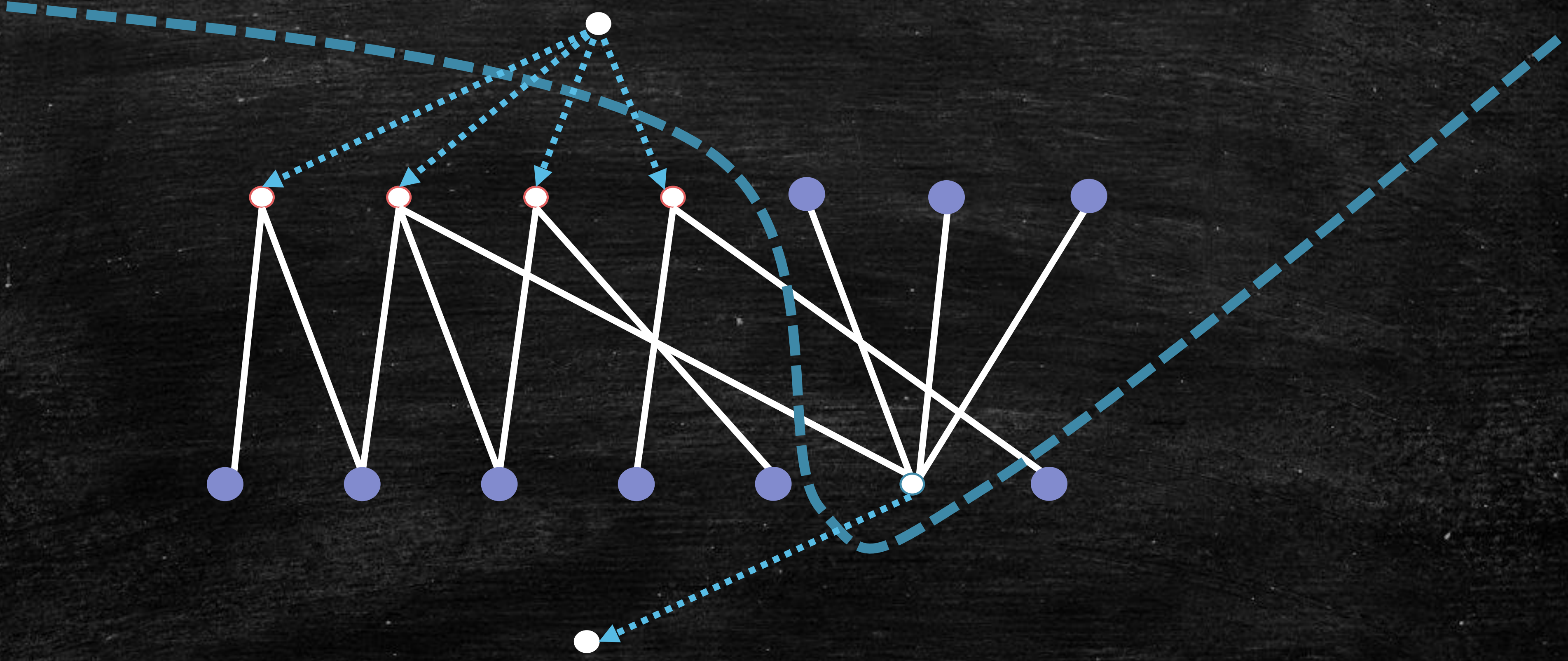


Min Vertex Cover = 5



The vertices being cut from s and t form a vertex cover.

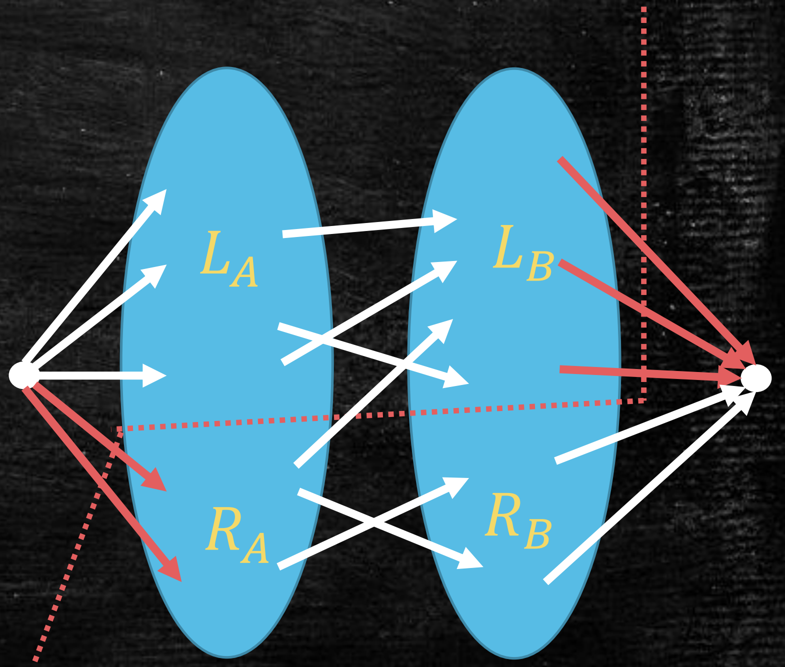
Max Independent Set = 9 (14 – 5 = 9)



The remaining vertices form an independent set.

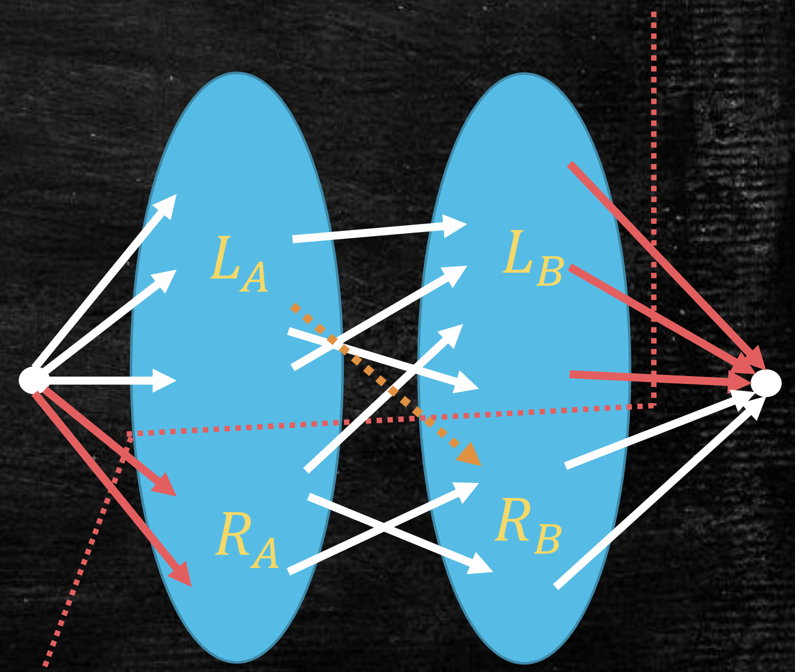
Max Independent Set/Min Vertex Cover

- $R_A \cup L_B$ is a vertex cover
- $L_A \cup R_B$ is an independent set
- Why these are true?



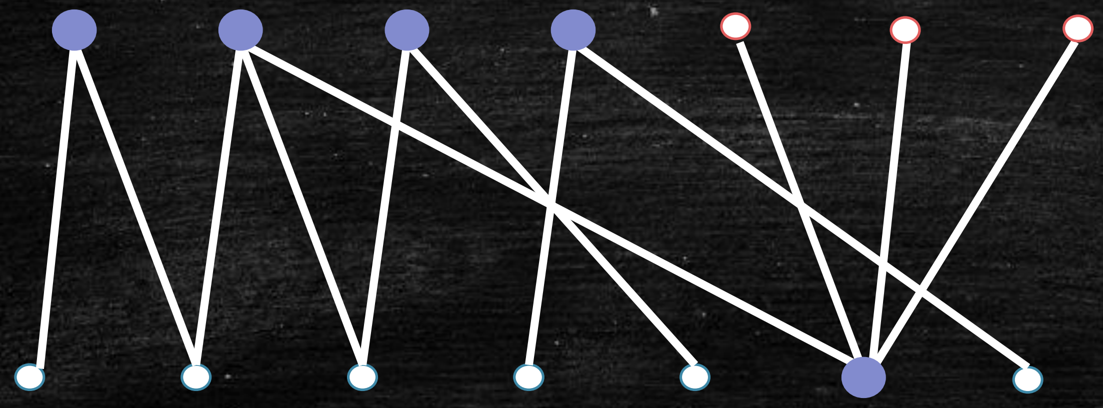
Max Independent Set/Min Vertex Cover

- $R_A \cup L_B$ is a vertex cover
- $L_A \cup R_B$ is an independent set
- Why these are true?
- Observation: No edge from L_A to R_B
  - O.w., the cut has size ∞, cannot be minimum



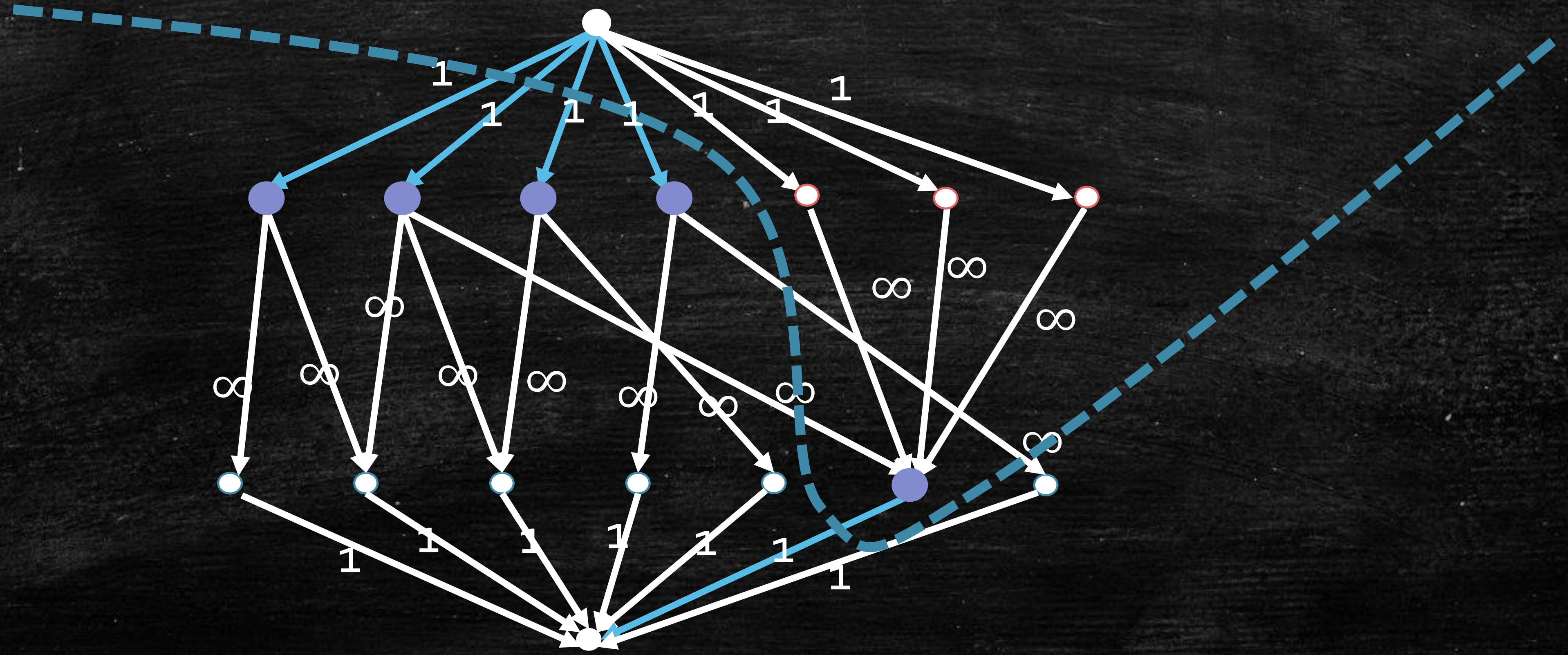
The Opposite Direction

- Given a vertex cover…



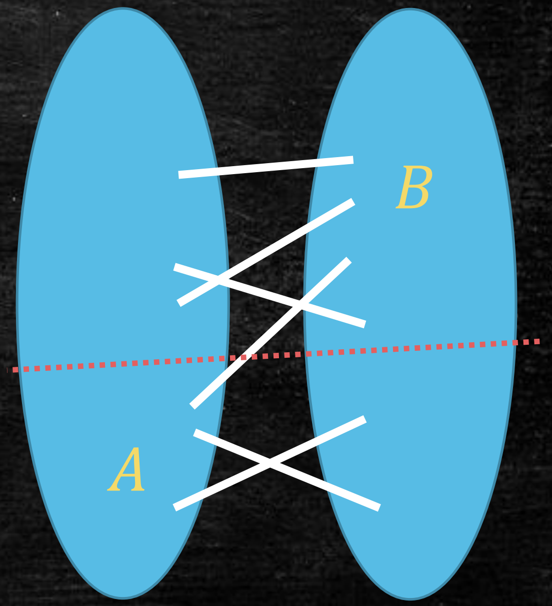
The Opposite Direction

- Given a vertex cover…
- Can we say that the blue edges define a cut?



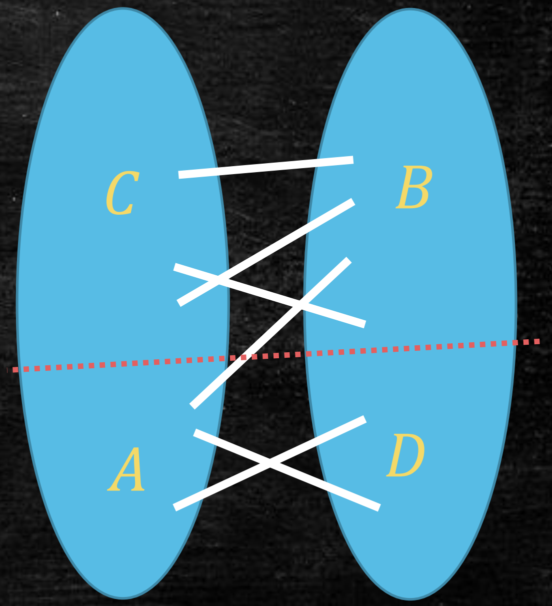
The Opposite Direction

- Suppose $A \cup B$ is a vertex cover



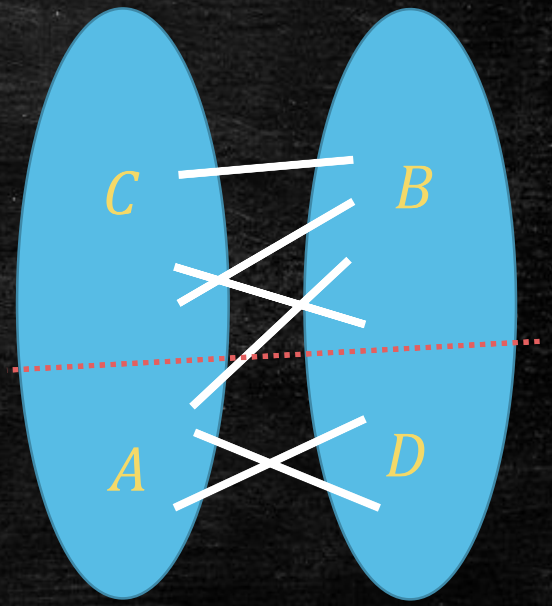
The Opposite Direction

- Suppose $A \cup B$ is a vertex cover
- Let C/D be those remaining vertices on the left/right.



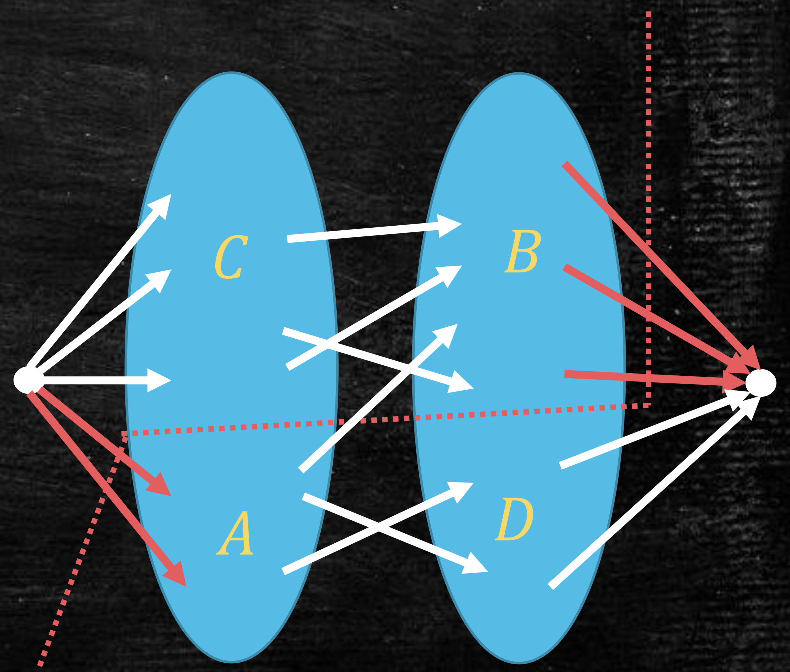
The Opposite Direction

- Suppose $A \cup B$ is a vertex cover
- Let C/D be those remaining vertices on the left/right.
- No edge from C to D.



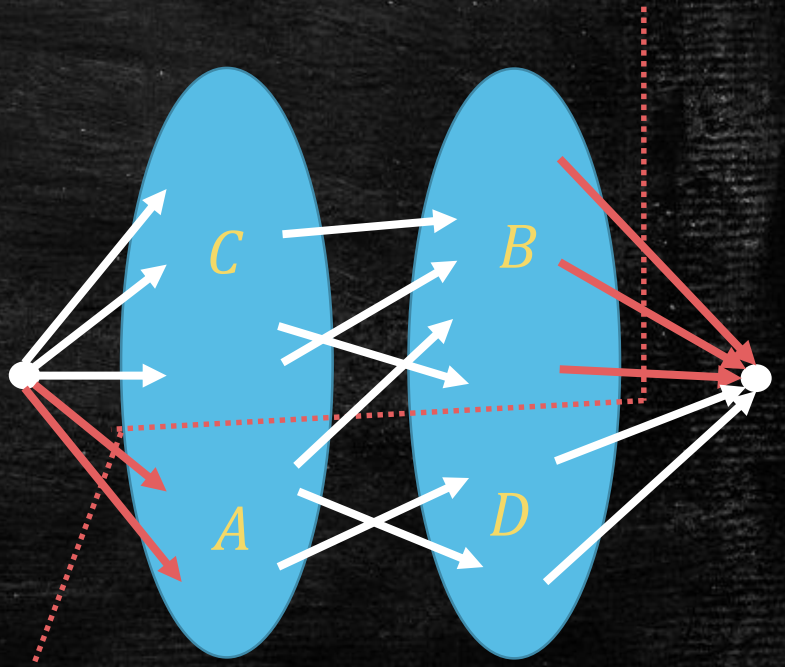
The Opposite Direction

- Suppose $A \cup B$ is a vertex cover
- Let C/D be those remaining vertices on the left/right.
- No edge from C to D.
- $(\{s\} \cup C \cup B,\ A \cup D \cup \{t\})$ is a cut with a finite size



The Opposite Direction

- Suppose $A \cup B$ is a vertex cover
- Let C/D be those remaining vertices on the left/right.
- No edge from C to D.
- $(\{s\} \cup C \cup B,\ A \cup D \cup \{t\})$ is a cut with a finite size
- and its size is $|A| + |B|$



Putting Two Directions Together

- There is a one-to-one correspondence between a vertex cover and a cut.
- Finding a minimum vertex cover is equivalent as finding a minimum cut.
- Can you fill in the remaining details for the followings?
  - An algorithm to find a minimum vertex cover on bipartite graphs
  - An algorithm to find a maximum independent set on bipartite graphs

Applications of Matching

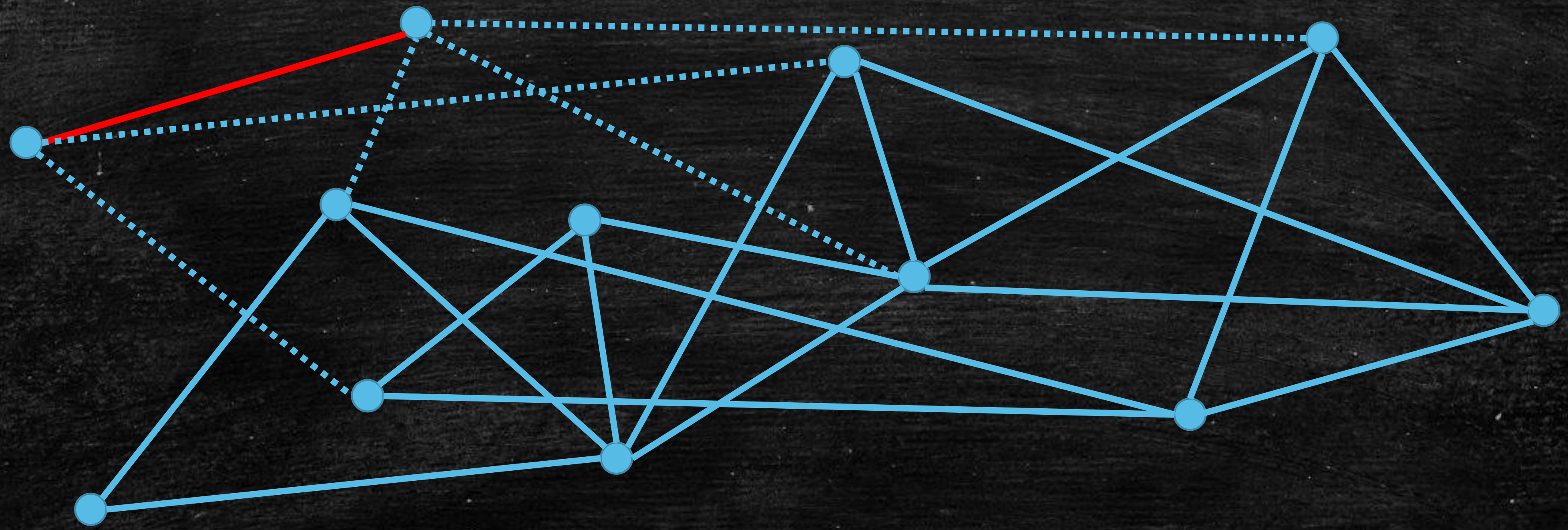
Approximation Algorithm for Vertex Cover

Minimum Vertex Cover

- Minimum Vertex Cover on general graphs is NP-hard.
- We will design a 2-approximation algorithm based on maximal matching.
- A matching M is maximal if no more edge can be added to M while still forming a matching.
- A simple greedy algorithm finds a maximal matching.

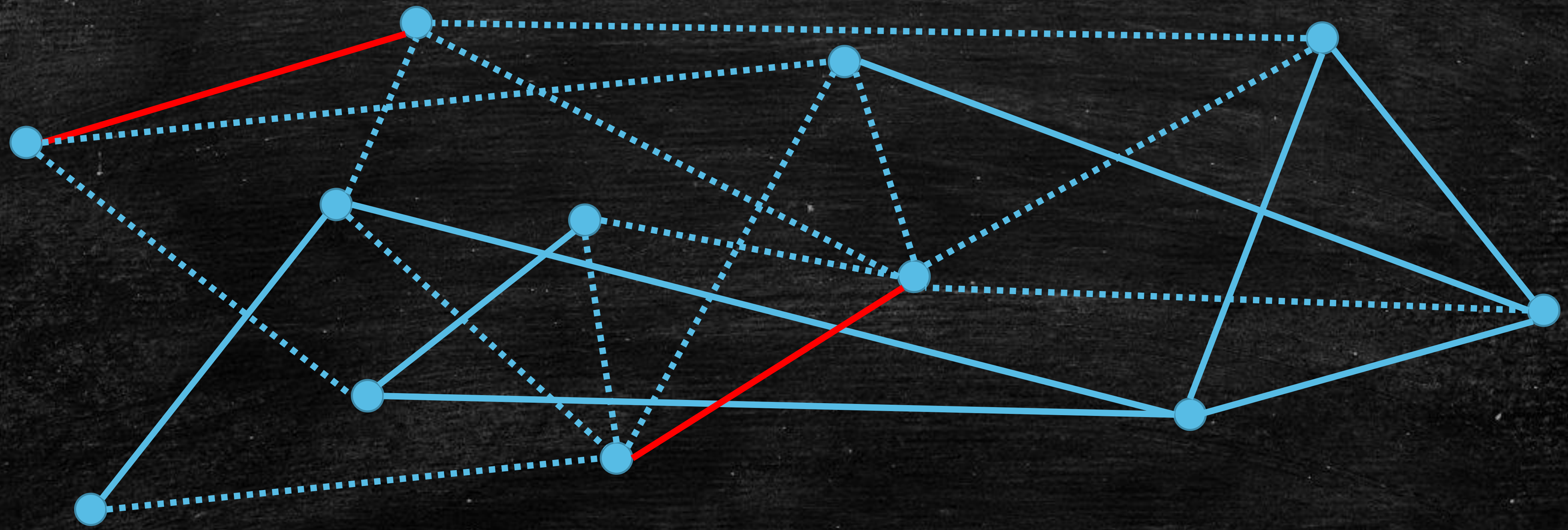
Finding a maximal matching

- Iteratively add an edge until no more edges can be added!



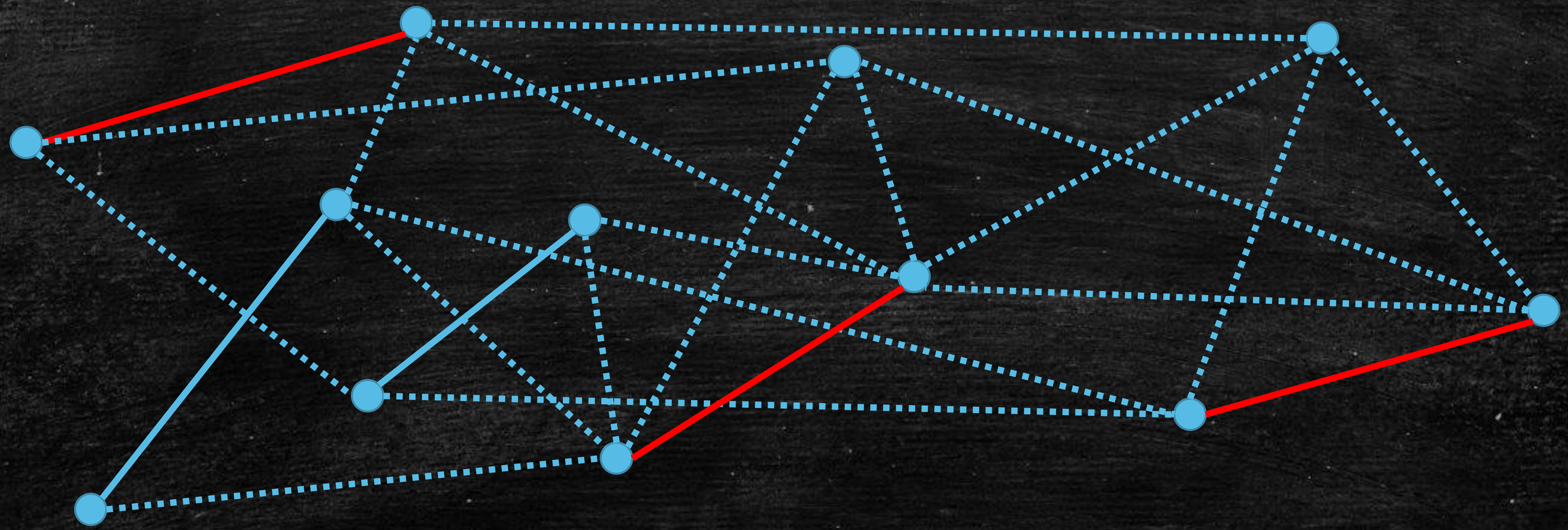
Finding a maximal matching

- Iteratively add an edge until no more edges can be added!



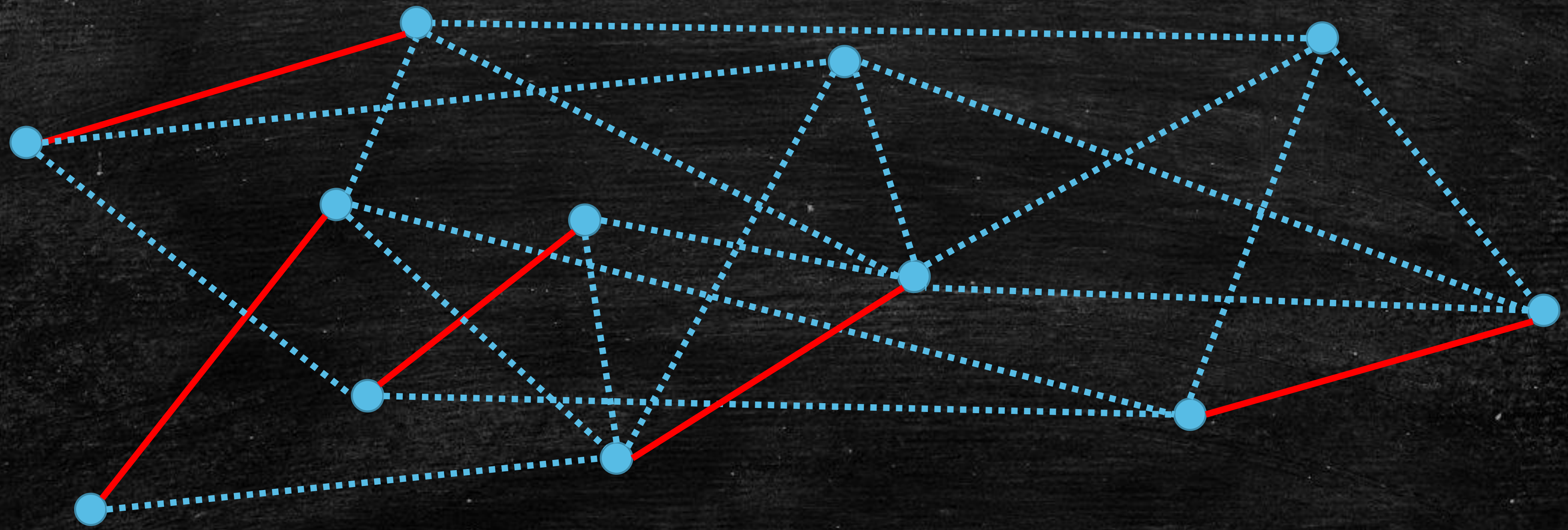
Finding a maximal matching

- Iteratively add an edge until no more edges can be added!



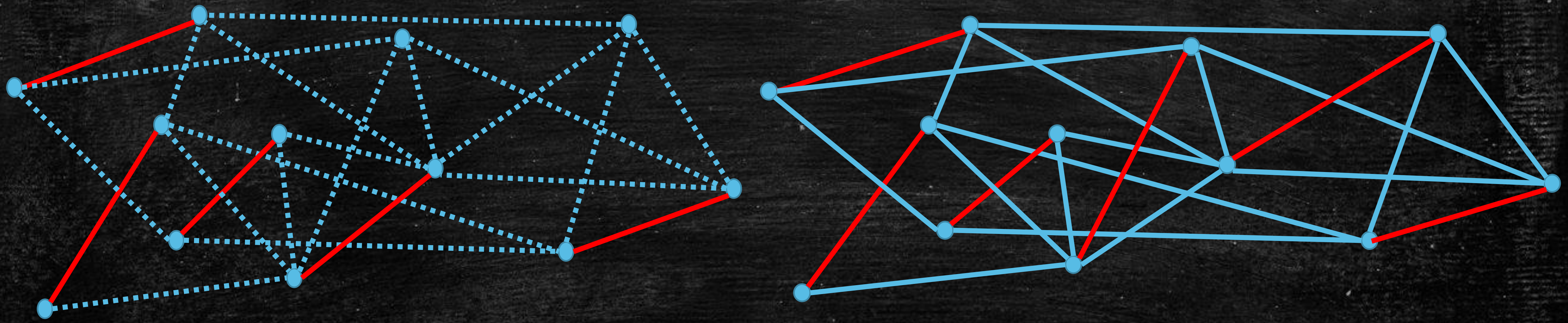
Finding a maximal matching

- Iteratively add an edge until no more edges can be added!

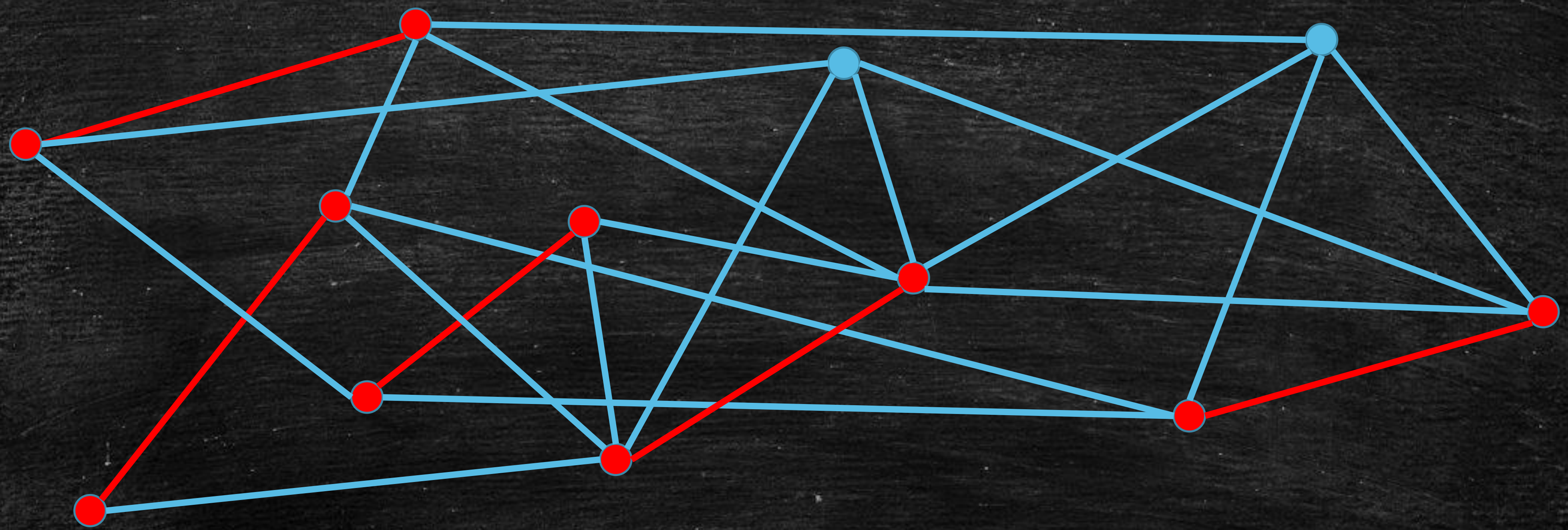


Maximal vs Maximum

- A maximal matching may not be maximum!



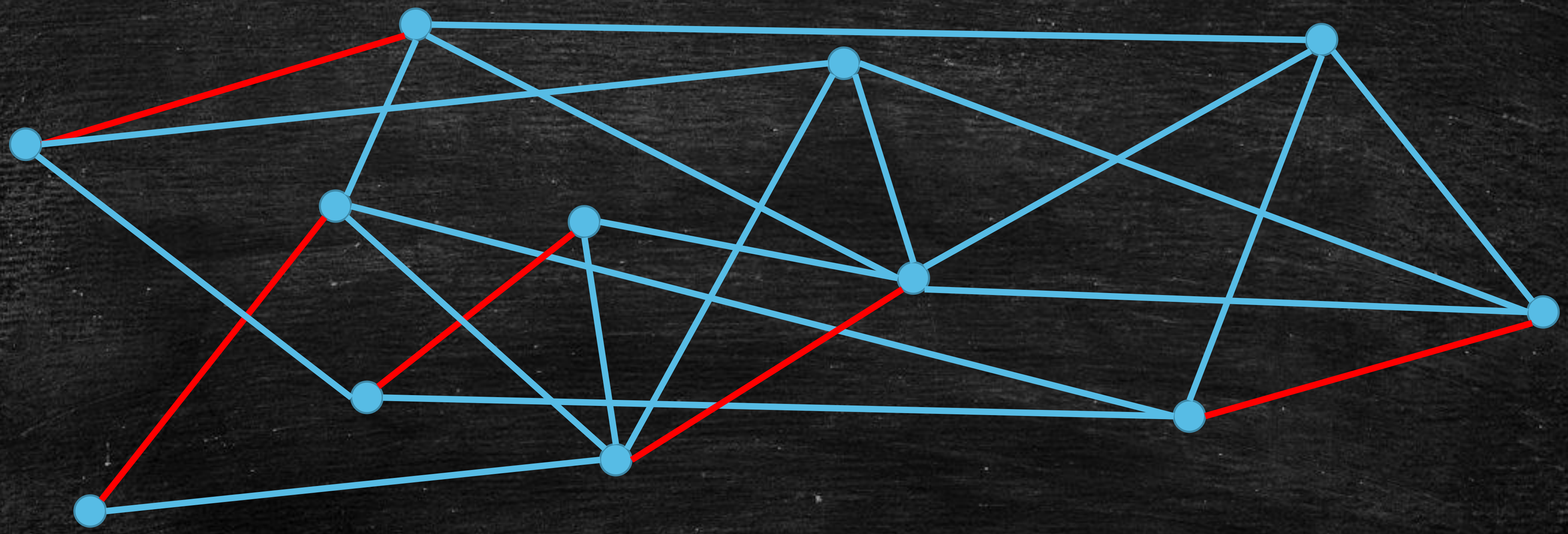
Lemma 1. The set of endpoints for all edges in a maximal matching is a vertex cover.



Proof. Let $M \subseteq E$ be a maximal matching.

- For any edge $e = (u, v)$, one or both of u, v must be an endpoint of an edge in M. (Otherwise, $M \cup \{e\}$ is still a matching, and M is not maximal.)
- This already implies endpoints of M is a vertex cover!

Lemma 2. For any maximal matching M, the size of any vertex cover is at least $|M|$.



Proof.

- Edges in M must be covered
- A vertex cannot cover two edges in M
- We need $|M|$ vertices to at least cover edges in M

A 2-approximation algorithm

Algorithm 1:

- Find a maximal matching M
- Let S be the endpoints of all edges in M
- Output S

Given an undirected graph $G = (V, E)$, let

- $OPT(G)$ be the size of a minimum vertex cover
- $S(G)$ be the vertex set output by Algorithm 1

Theorem: *For any undirected graph G, we have* $|S(G)| \leq 2 \cdot OPT(G)$

$\forall G: |S(G)| \leq 2 \cdot OPT(G)$

- Lemma 1. *The set of endpoints for all edges in a maximal matching is a vertex cover.*
- $\Rightarrow S(G)$ is a vertex cover
- $|S(G)| = 2|M|$
- Lemma 2: *For any maximal matching M, the size of any vertex cover is at least $|M|$.*
- $\Rightarrow OPT(G) \geq |M|$

Revisiting our 2-approximation algorithm

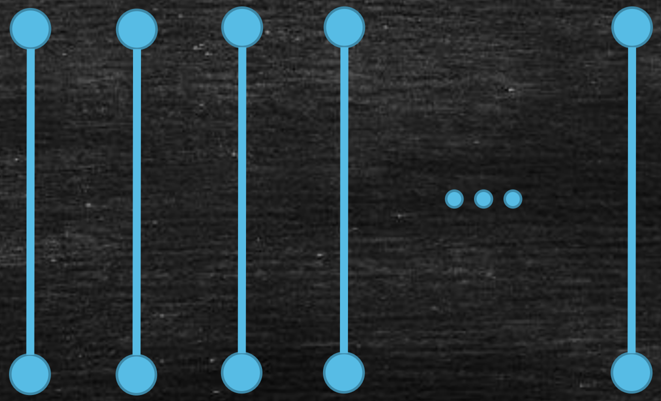
<u>Algorithm 1</u>:

- Find a maximal matching M
- Let S be the endpoints of all edges in M
- Output S

Question: Can we do better than 2-approximation?

- Idea 1: same algorithm with a more careful analysis?
- Idea 2: another more clever algorithm?

Idea 1 doesn't work

$G =$ 

- Suppose G has $2n$ vertices and n edges as above.
- $OPT(G) = n$
- $\mathcal{A}(G) = 2n$

Idea 2 is unlikely to work

- [Khot & Regev, 2008] Assuming Unique Game Conjecture, if minimum vertex cover has a polynomial time $(2 - \epsilon)$-approximation algorithm for some $\epsilon > 0$, then $\mathbf{P} = \mathbf{NP}$.
- [Khot, Minzer & Safra, 2017] If minimum vertex cover has a polynomial time $(\sqrt{2} - \epsilon)$-approximation algorithm for some $\epsilon > 0$, then $\mathbf{P} = \mathbf{NP}$.

Today's Lecture

- Edmonds-Karp Algorithm
- Applications of Max-Flow to assignment-styled problems
  - Dinner Table Assignments
  - Tournament
- Max-Flow and Matching
- Min-Cut and Max Independent Set/Min Vertex Cover
- A 2-approximation algorithm for min vertex cover based on maximal matching