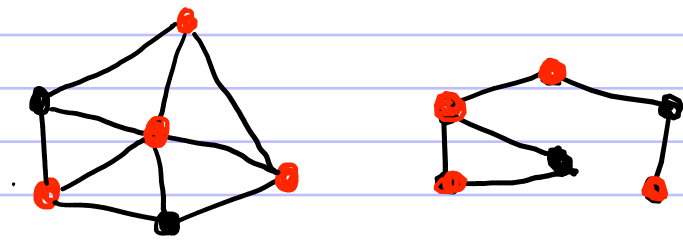
# MATCHING IN BIPARTITE GRAPHS

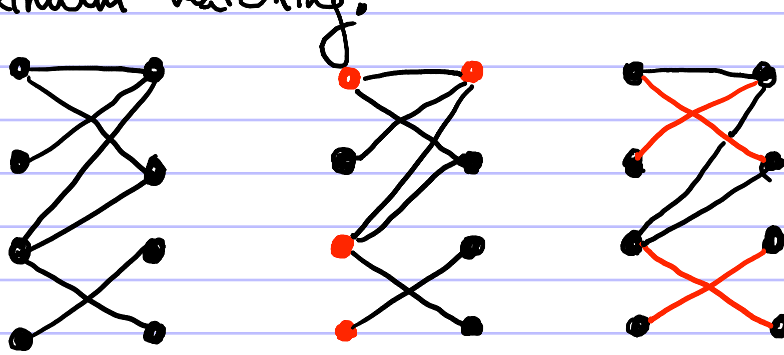
A vertex cover of a graph $G=(V,E)$ is a subset $V' \subseteq V$ such that $\forall \{v_1, v_2\} \in E$, $v_1$ or $v_2$ is in $V'$.



{•} is a cover

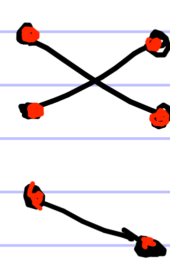
Minimum vertex cover problem: Given a graph $G$, find a smallest vertex cover of $G$.

König's Theorem: For any bipartite graph $G=(V_1, V_2, E)$, the size of any smallest vertex cover equals the size of any maximum matching.



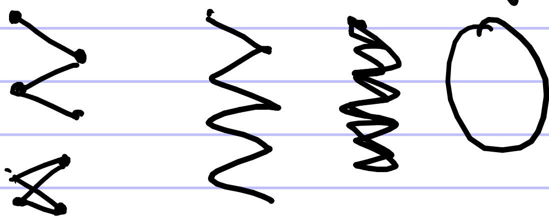

Let $\alpha(G)$ be size of maximum matching, $\beta(G)$ be size of min vertex cover.

It is clear $\alpha(G) \leq \beta(G)$. Why?



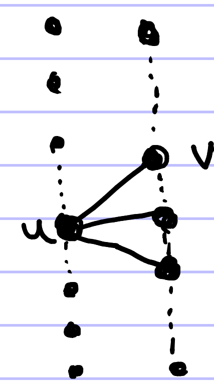
So the tough part is showing $\alpha(G) \ge \beta(G)$.

Let $G$ be a minimal bipartite graph with $\alpha(G) \ne \beta(G)$.
So $G$ is connected and has degree $> 2$. Why?



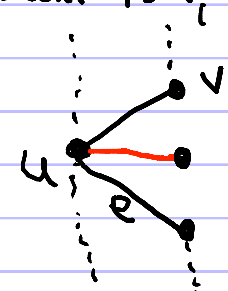
So $G$ has a vertex $u$ of degree-3 or higher with neighbor $v$:

If $\alpha(G-v) < \alpha(G)$, then since $G$ is minimal, $G-v$ has vertex cover $V'$, with $|V'| = \alpha(G-v) < \alpha(G)$.
Then $V' \cup \{v\}$ is a cover for $G$ and so $\beta(G) = \alpha(G)$. $\rightarrow\leftarrow$ Contradiction.
So $\alpha(G-v) = \alpha(G)$.




So there is a max matching $E'$ of $G$ with no edges incident to $v$.

Let $e$ be an edge incident to $u$, not incident to $v$, not in $E'$:
Let $V''$ be a cover of $G-e$ with $|V''| = |E'| = \alpha(G)$.
Since $E'$ has no edges incident to $v$, and $V''$ must have a vertex on every edge in $E'$, $v \notin V''$. So $u \in V''$.

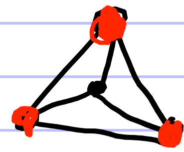



So $V''$ is a cover for $G$ as well. So $\alpha(G) = |V''| \ge \beta(G)$.

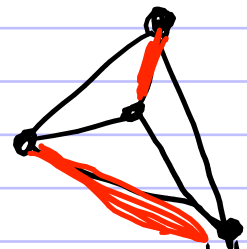
So $\alpha(G) \le \beta(G)$ and $\beta(G) \le \alpha(G)$. So $\alpha(G) = \beta(G)$.

Hall = König = Egerváry = Menger = Dilworth = Birkhoff-VonNeumann

# Does König's Theorem generalize to non-bipartite graphs?

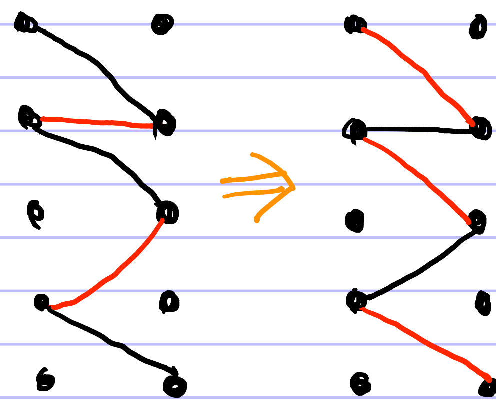


min VC = 3

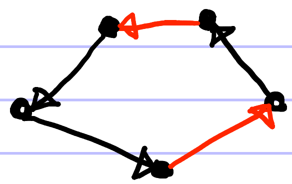


max match = 2

Recall alternating paths approach to finding a maximum matching in a bipartite graph:



What could go wrong with this in a general graph?



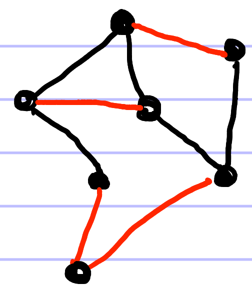
Odd cycles!
(The only difference between bipartite & general graphs)

Idea: When looking for an alternating path, if an odd cycle is encountered, contract the cycle into a "supernode" and run BFS again on new graph.

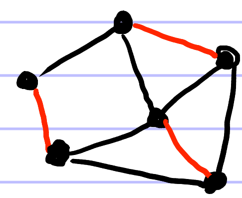
The maximum matching problem for general graphs has a polynomial-time algorithm (<u>Edmonds' Blossom Algorithm</u>).

# MINIMUM EDGE COVERS

An edge cover of a graph $G=(V,E)$ is a subset $E' \subseteq E$ such that every vertex in $V$ is an endpoint of some edge in $E'$.

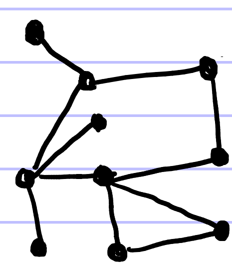


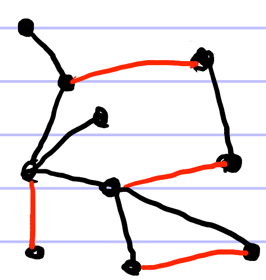
Edge cover



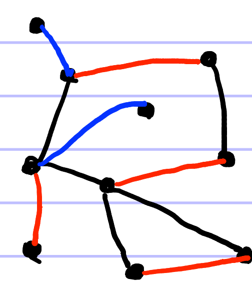
Perfect matching
(and an edge cover)

Homework: Prove perfect matchings are always edge covers.





Max matching



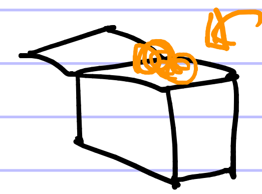
Min edge cover

Homework: Prove any maximum matching can be augmented into a minimum edge cover by adding one edge per uncovered vertex.

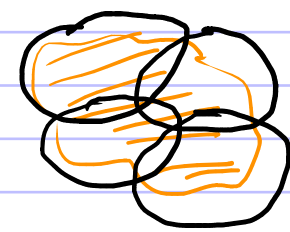
# PACKING AND COVERING IN GRAPHS

| Problem | Type |
|---|---|
| Matching | Packing |
| Vertex cover | Covering |
| Edge cover | Covering |
| Independent set | Packing |
| Dominating set | Covering |

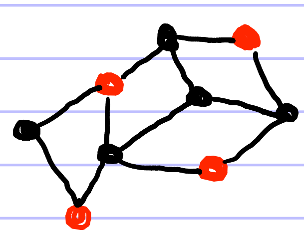
PACKING is MAXIMIZATION

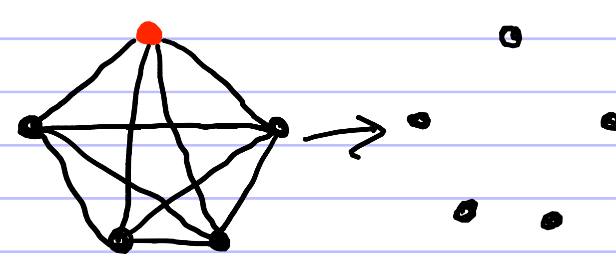


COVERING is MINIMIZATION



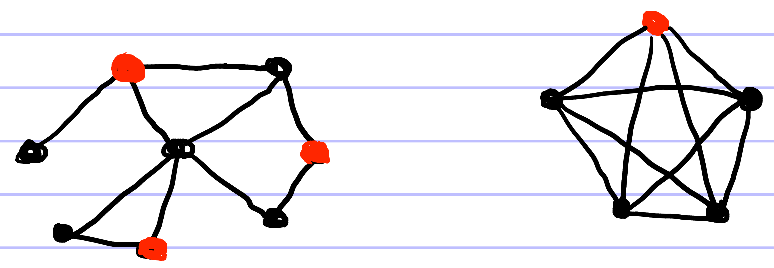
An independent set of a graph $G=(V,E)$ is a subset $V' \subseteq V$ such that for every edge $\{v_1, v_2\} \in E$, either $v_1 \notin V'$ or $v_2 \notin V'$.





clique $= K_n$ for some $n$ $\subseteq G$

A dominating set of a graph $G=(V,E)$ is a subset $V' \subseteq V$ such that every vertex in $V$ is in $V'$ or is a neighbor of a vertex in $V'$



Vertex cover: cover edges w/vertices.
Dominating set: cover vertices w/vertices.

| Polynomial-time algorithms | Unknown |
|---|---|
| Edge cover | Vertex cover |
| Matching | Independent set |
| Vertex cover in bipartite graphs | Dominating set |

Maybe some problems don't have polynomial-time algorithms? Some definitely don't by pidgeonhole.

But what about ... vertex cover ?
indep. set
⋮

Restrict our problems to "YES"/"NO" outputs.

Ex: Given a graph G and integer a, does G have a matching of size at least a? (Maximum matching problem)

A polynomial-time many-one reduction from Problem 1 to Problem 2 is a polynomial-time algorithm that does the following:

Input: An instance A of problem 1

Output: An instance B of problem 2

Also: The answer to A is "YES" iff the answer to B is "YES"

Example:

Problem 1: Given a graph G and integer k, does G have an independent set of size $\geq b$?

Problem 2: Given a graph G and integer b, does G have a vertex cover of size $\leq b$?

Reduction: Given $\langle (V,E), b \rangle$, return $\langle (V,E), |V|-b \rangle$.

Lemma: A graph $G=(V,E)$ has an independent set $V' \subseteq V$ iff $V-V'$ is a vertex cover.

Pf: If $V'$ is an independent set, $V-V'$ has at least one endpoint of every set. If $V-V'$ is a vertex cover, $V'$ is missing at least one endpoint of every edge.

By Lemma, the answer to $\langle (V,E), b \rangle$ is "Yes" iff the answer to $\langle (V,E), |V|-b \rangle$ is "Yes".
So the reduction works.

# NP-hardness

A problem P is NP-hard if there exists a polynomial-time many-one reduction from another NP-hard problem to P.

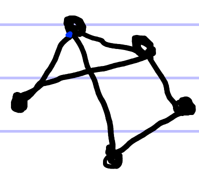
Theorem: Independent set is NP-hard.

Corollary (from our reduction & this theorem): vertex cover is NP-hard.

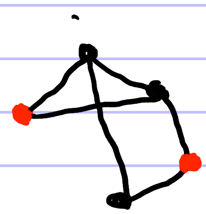
---

NP-hardness ≈ as hard as any problem in the complexity class NP.

The complexity class NP is the set of problems whose proposed solutions can be verified by a polynomial-time algorithm.



$\langle G, a \rangle$
Independent set



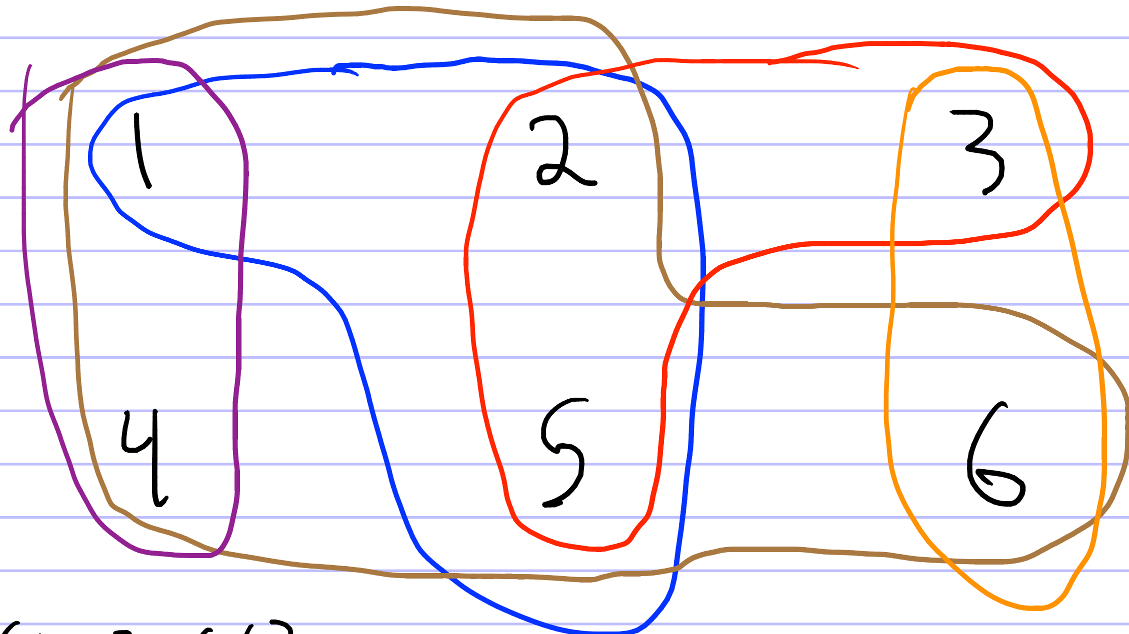
proposed solution

Conjecture: The complexity class NP contains problems that do not have polynomial-time algorithms.

Corollary: NP-hard problems do not have polynomial-time algorithms.

# SET COVER

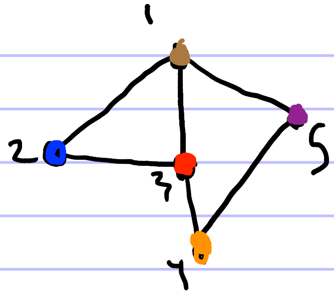
**Set cover problem:** Given a set $U = \{1, 2, \ldots, n\}$, a set of sets $S = \{S_1, S_2, \ldots, S_m\}$ with $S_i \subseteq U$, and an integer $b$, does there exist a subset $S' \subseteq S$ with $\bigcup_{S_i \in S'} S_i = U$ such that $|S'| \leq b$?



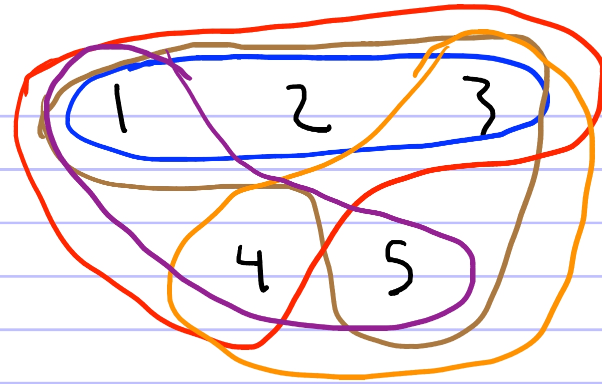

$U = \{1, 2, 3, 4, 5, 6\}$

$S = \{\{1, 4\}, \{1, 2, 5\}, \{3, 6\}, \{2, 3, 5\}, \{1, 2, 4, 5, 6\}\}$

Claim: Dominating set problem and set cover problem have polynomial-time many-one reductions to each other.



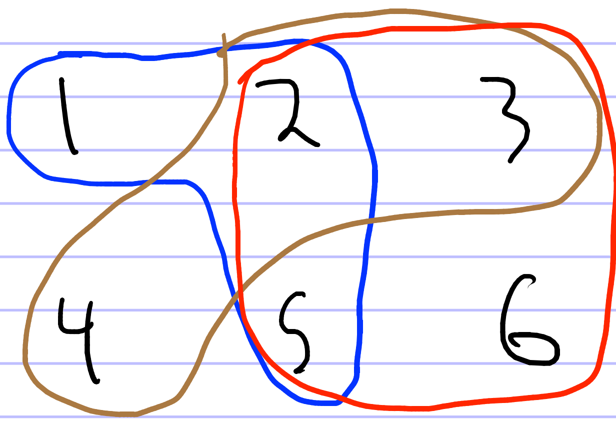

G

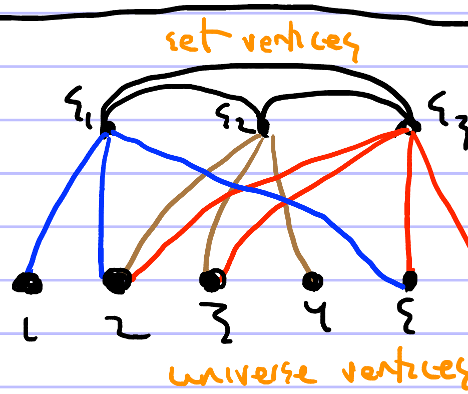



$\langle G, a\rangle \longrightarrow \langle U, S, a\rangle$

Reduction <u>from</u> dominating set <u>to</u> set cover.

---




$\langle U, S, a\rangle \longrightarrow \langle G, a\rangle$

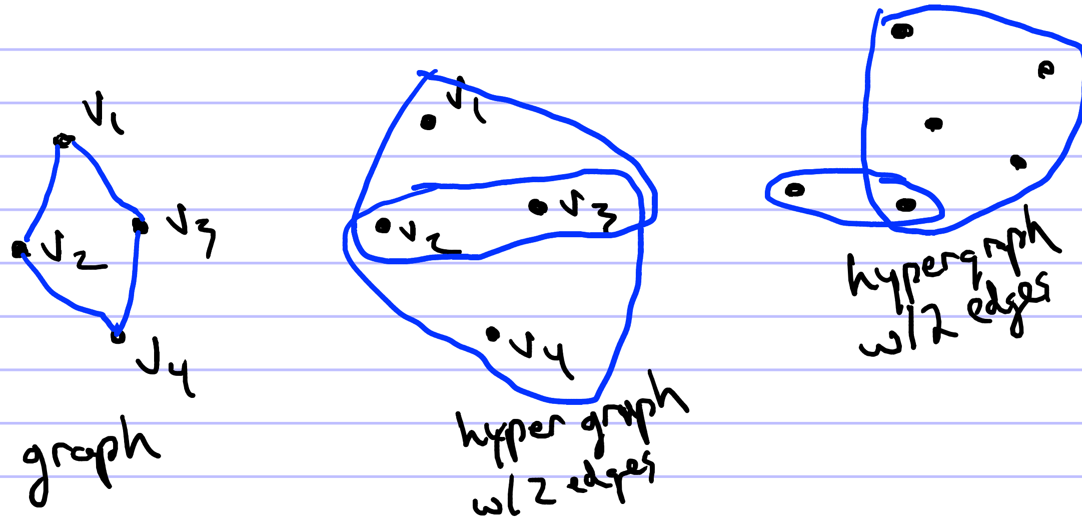
Reduction <u>from</u> set cover <u>to</u> dominating set.

If there are $\leq a$ sets that cover, then choose the set vertices corresponding to these sets.

If there are $\leq a$ vertices that dominate G, first swap any universe vertices with some set vertex adjacent to them. This set of set vertices corresponds to a set cover.

# HYPERGRAPHS

A hypergraph $G=(V,E)$ consists of a set of vertices $V$ and a set of hyperedges $E$. Each hyperedge $e \in E$ is a subset $e \subseteq V$.

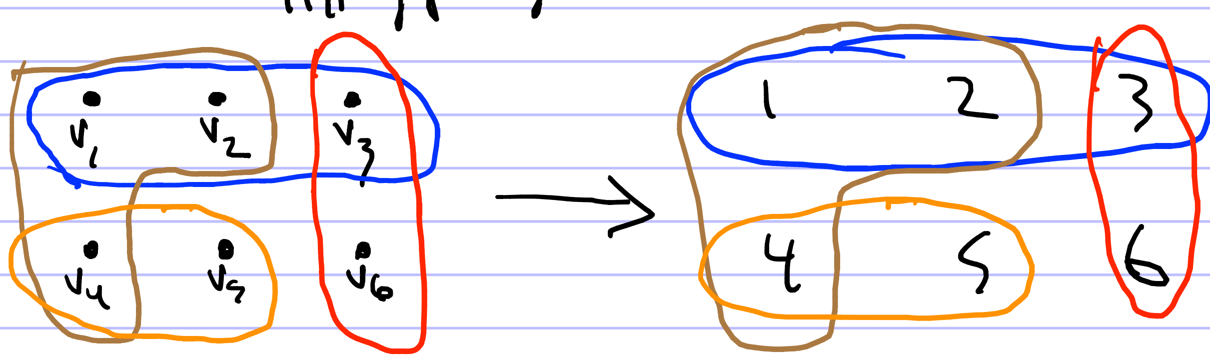



A hypergraph $G=(V,E)$ is $k$-uniform if for all edges $e \in E$, $|e|=k$.

All graphs are 2-uniform hypergraphs.

Hypergraph edge cover and vertex cover problems are defined analogously: find a small set of hyperedges or vertices covering the graph.

Hypergraph edge cover $\approx$ set cover.




Hypergraph with 6 vertices, 4 hyperedges.

Set cover instance with $U=\{1,2,3,4,5,6\}$, $S=\{\{1,2,3\},\{1,2,4\},\{3,6\},\{4,5\}\}$

Theorem: the vertex cover problem constrained to 3-regular graphs is NP-hard.

Homework: Use this theorem to show that the hyperedge cover problem constrained to 3-uniform hypergraphs is NP-hard.

2-uniform has polytime algorithm

3-uniform is NP-hard