Hardness of computational problems.

Hamiltonian cycle: Simple cycle that contains all vertices.

$HC := \{ G \mid G \text{ has a Hamiltonian cycle} \}$.

Given a graph H, an "easy" way to check if it has Ham. Cycle is to check if $H \in HC$.

$\underline{\underline{\{0,1\}^{\binom{n}{2}}}}$

Each string encodes a graph on n vertices.

$\{0,1\}^{\binom{n}{2}}$

Those that are in HC   Rest.

$f_{HC} : \{0,1\}^{\binom{n}{2}} \longrightarrow \{0,1\}$ tells us if given instance $\in HC$ or not.

One way to generate f_{HC} is

- Enumerate all n-vertex and compute in each graph (by brute force) if it has a Ham. cycle
- Build a truth table.
- Construct a fn from truth table.

$2^{\binom{n}{2}}$

1	2	...	$\binom{n}{2}$	HC
0	1			

Thm: Maximum Independent Set is NP-hard.

↳ maximal Independent Set has $O(m+n)$ greedy alg.

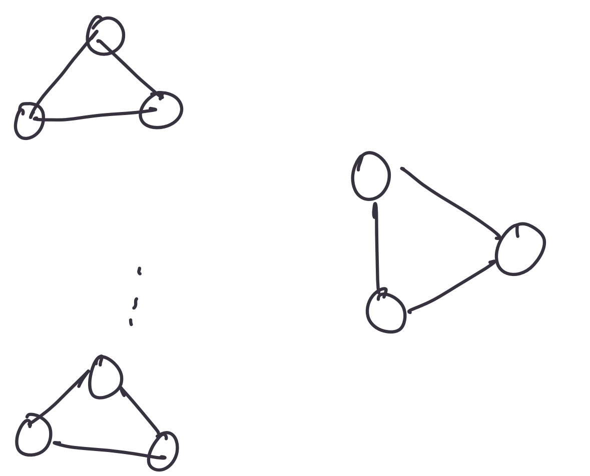
Independent set: A subset of vertices $I \subseteq V$ s.t for any pair of vertices in I, they do not have an edge connecting them.

→ Want: Ind set of maximum cardinality.

Maximal Independent Set: Ind. set s.t adding any more vertices ~~could~~ will introduce an edge amongst a pair of vertices in I.

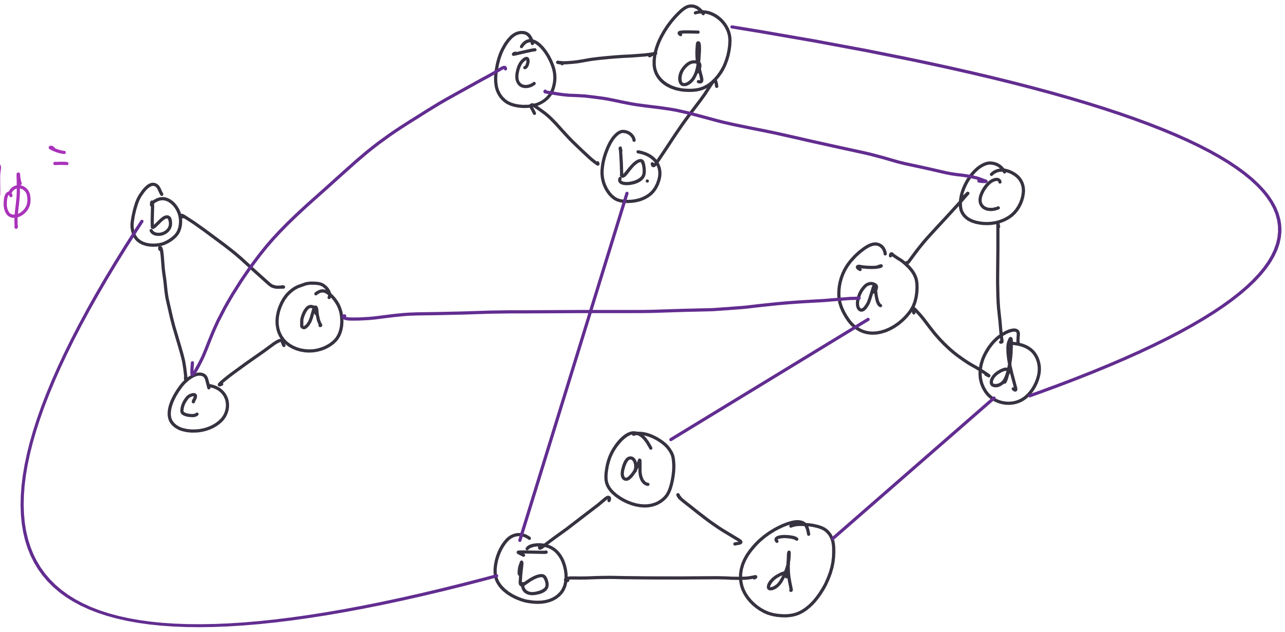
$3SAT \leq_p$ Maximum IS.

$$\underbrace{C_1 \wedge C_2 \wedge \cdots \wedge C_m}$$



$\phi = (a \vee b \vee c) \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b} \vee \bar{d})$

$G_\phi =$



Claim: ϕ is sat. if and only if the graph G_ϕ has a Max IS of size m.

$(\Rightarrow)$
Take the sat. assignment. Then pick one literal from each clause that is set to true. The corr. vertices in G_ϕ are an IS.

$(\Leftarrow)$ Pick vertices from MIS set them to true. That gives a sat assignment.

$C \wedge \bar{C}$

$a \qquad \bar{a}$

Reductions and Completeness

— Garey and Johnson $\Big\}$ (Non-exhaustive) Collection of NP-complete problems and reductions

Vertex cover: Set of vertices such that for every edge one of its end points is in the set.

$\underline{\text{Qn}}$: Is there a Vertex Cover of size at most k?

Brute force $\rightarrow$ $\underbrace{\binom{n}{k}}_{\sim\left(\frac{en}{k}\right)^k} \cdot m.$ $\longrightarrow$ $\underline{\underline{n^k \cdot \text{poly}(n)}}$.

$n^{\log n}$

↳ Can we get an algorithm that runs in time

$\underline{2^{f(k)} \cdot \text{poly}(n)}$. $\longrightarrow$ $n \cdot \text{poly}(n)$

(FPT)

Fixed Parameter Tractable algorithms