NP-Hardness

How to prove a problem L is hard?
↑ can't be solved in polytime

idea - by reductions

To show: if L could be solved in polytime,
then some well-known hard problem L_0
could be solved in polytime
contradicting hypothesis that L_0 can't be
solved in polytime

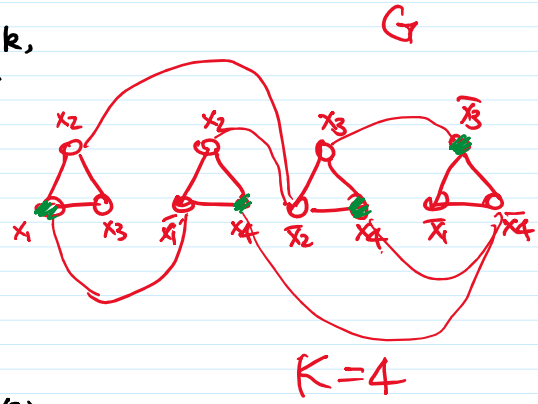
What choice of L_0?

Problem (SAT) Given Boolean formula F ← in CNF
in n vars $x_1, \dots, x_n$.
decide whether ∃ assignment of $x_1 \dots x_n$
s.t. $F(x_1, \dots, x_n)$ is true

if all clauses have length k, ⇒ k-SAT

e.g. $F = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$
(clauses; literal)

set $x_4 = 1$, $x_1 = 1$, $x_3 = 0$ ←



brute force: $\tilde{O}(2^n)$ time

3-SAT $T(n) = 7T(n-3) \Rightarrow O(7^{n/3})$

Schöning '99 $\tilde{O}((4/3)^n)$

(AI'26: $2^{O(n/\sqrt{\log n} \cdots)}$)

Hypothesis: SAT can't be solved in polytime.
(equiv. to "P ≠ NP")

Rmk: Why SAT?
most problems of the form ∃ ... s.t. ...
reduces to SAT
(Cook-Levin Thm: "NP")

"Definition" L is NP-hard iff SAT polytime-reduces to L.

L is NP-complete iff SAT is polytime-equiv to L.

to L.

L is NP-complete iff SAT is poly-time-equiv to L.
(i.e. can reduce in both dir.).

Ex1 SAT poly-time-reduces to 3SAT

rewrite $\alpha_1 \vee \cdots \vee \alpha_k$

as $(y_2 \to \alpha_1 \vee \alpha_2) \wedge (y_3 \to y_2 \vee \alpha_3) \wedge \cdots \wedge (y_k \to y_{k-1} \vee \alpha_k) \wedge y_k$

$\alpha \to \beta \equiv \bar{\alpha} \vee \beta$

$\equiv (\bar{y}_2 \vee \alpha_1 \vee \alpha_2) \wedge (\bar{y}_3 \vee y_2 \vee \alpha_3) \wedge \cdots \wedge (\bar{y}_k \vee y_{k-1} \vee \alpha_k) \wedge y_k$

Ex2 3SAT poly-time-reduces to 0-1 Integer Linear Prog. (ILP)

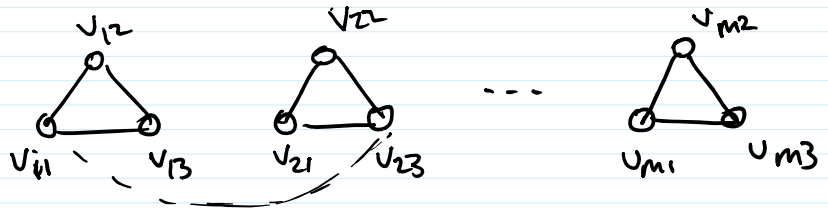
rewrite $x_1 \vee \bar{x}_2 \vee x_3$

as $x_1 + (1 - x_2) + x_3 \geq 1$.

Ex3 3SAT poly-time-reduces to Independent-Set

given graph $G=(V,E)$, & K,
decide $\exists\, I \subseteq V$ s.t. $|I| \geq K$,
$\forall u,v \in I \Rightarrow uv \notin E$.

Given 3CNF-formula F with n vars, m clauses,
construct graph $G=(V,E)$, & K,



for each clause $C_i = \alpha_{i1} \vee \alpha_{i2} \vee \alpha_{i3}$,

for each clause $C_i = \alpha_{i1} \vee \alpha_{i2} \vee \alpha_{i3}$,
create triangle $v_{i1}\, v_{i2}\, v_{i3}$

whenever $\alpha_{ij} = \overline{\alpha_{i'j'}}$, add edge $v_{ij}\, v_{i'j'}$

$K = m$.

Correctness Pf: $\exists$ assignment s.t. F evaluates to true
$\Leftrightarrow$ $\exists$ indep set for G of size $\geq K$.

- - -

Ex4 Indep Set polytime-reduces to Vertex Cover

Given graph $G=(V,E)$, & K,
decide $\exists S \subseteq V$ s.t. $|S| \leq K$
$\forall uv \in E, \Rightarrow u \in S$ or $v \in S$.
or both.
(equiv. $u \notin S$ and $v \notin S \Rightarrow uv \notin E$)
(S is a VC iff S^c is an IS)

just change
K to n-K.

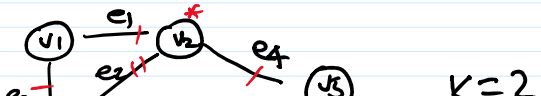
$p \to q$
$\bar{q} \to \bar{p}$

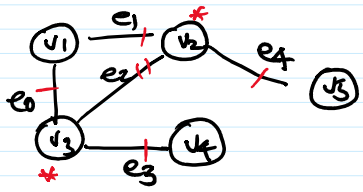
Set Cover
Vertex Cover → Subset Sum
Indep Set
0-1 ILP
3SAT
SAT

Ex5 Vertex Cover polytime-reduces to Subset-Sum

Given numbers $a_1, \ldots, a_n, W$,
decide $\exists S \subseteq \{a_1, \ldots, a_n\}$
that sums to W.

Given graph $G=(V,E)$, number K,





$K = 2$

construct numbers

incidence matrix

	e_4	e_3	e_2	e_1	e_0	
v_1	1	0	0	0	1	$= a_1$
* v_2	1	1	0	1	1	$= a_2$ ←
* v_3	1	0	1	1	0	$= a_3$ ←
v_4	1	0	1	0	0	$= a_4$
v_5	1	1	0	0	0	$= a_5$
					1	$= b_0$ ←
				1	0	$= b_1$ ←
			1	0	0	$= b_2$
		1	0	0	0	$= b_3$ ←
	1	0	0	0	0	$= b_4$ ←
	K	2	2	2	2	$= W$