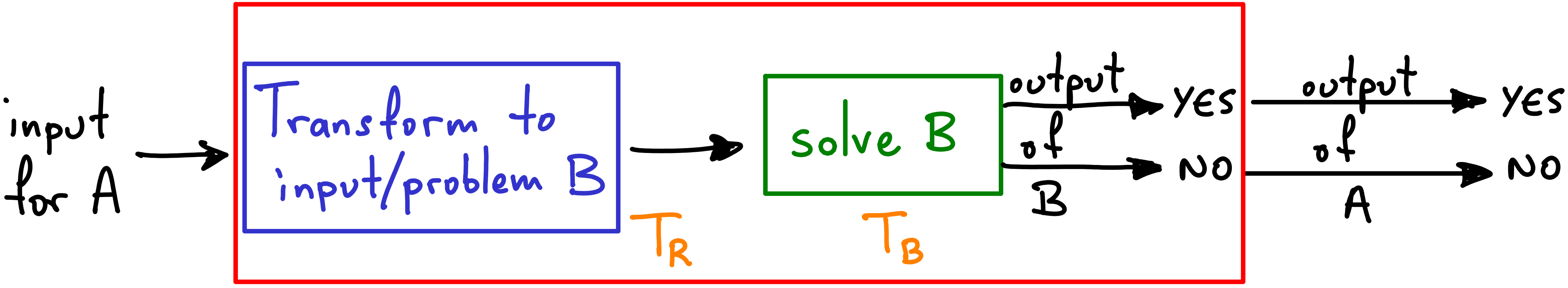
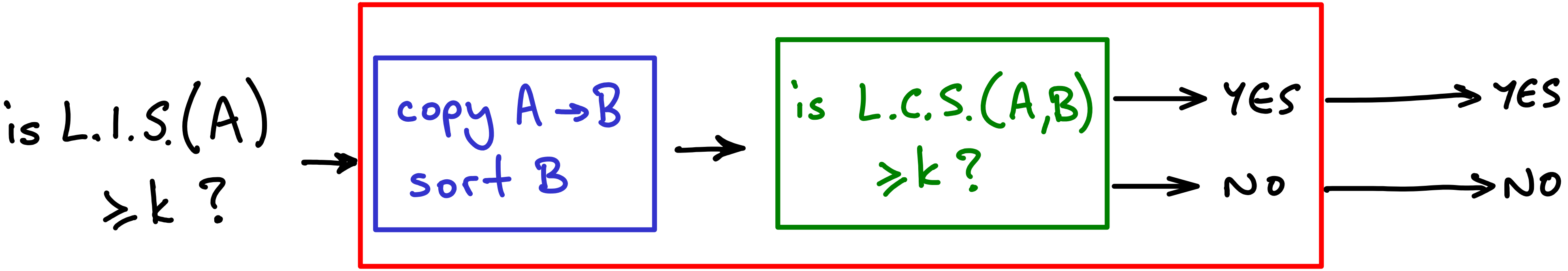
REDUCING ONE (DECISION) PROBLEM TO ANOTHER

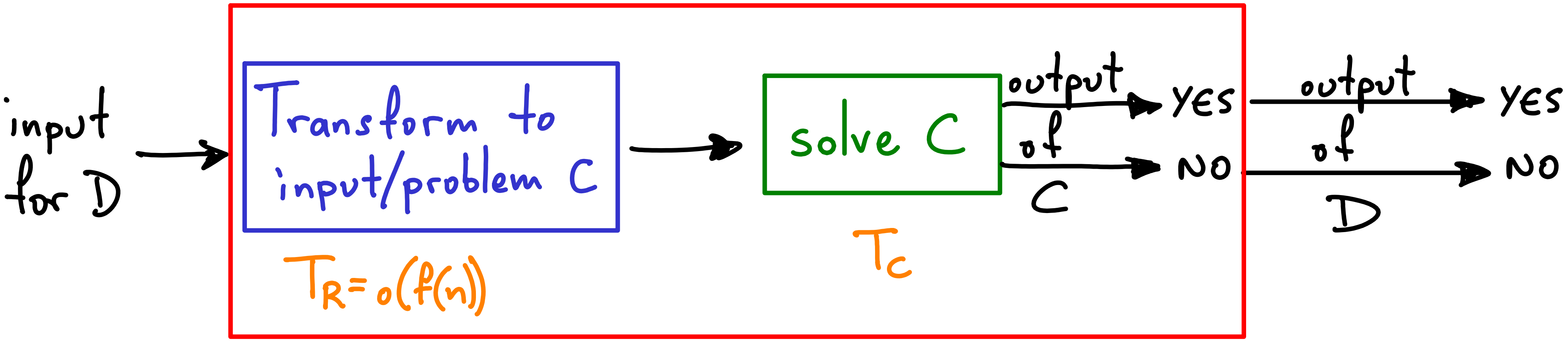
A B



solve A using B : $T_R + T_B$



LOWER BOUND FOR ONE (DECISION) PROBLEM VIA ANOTHER
C D



solve D: known $\Omega(f(n)) = T_R + T_C \Rightarrow T_C = \Omega(f(n))$

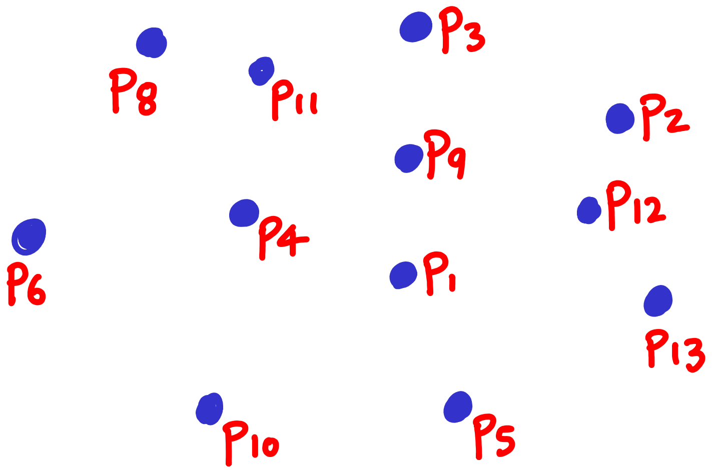
- quick transformation T_R $\Rightarrow$ C is at least as difficult as D

(fast solution for C $\Rightarrow$ fast solution for D)

The CONVEX HULL problem (definition by picture)

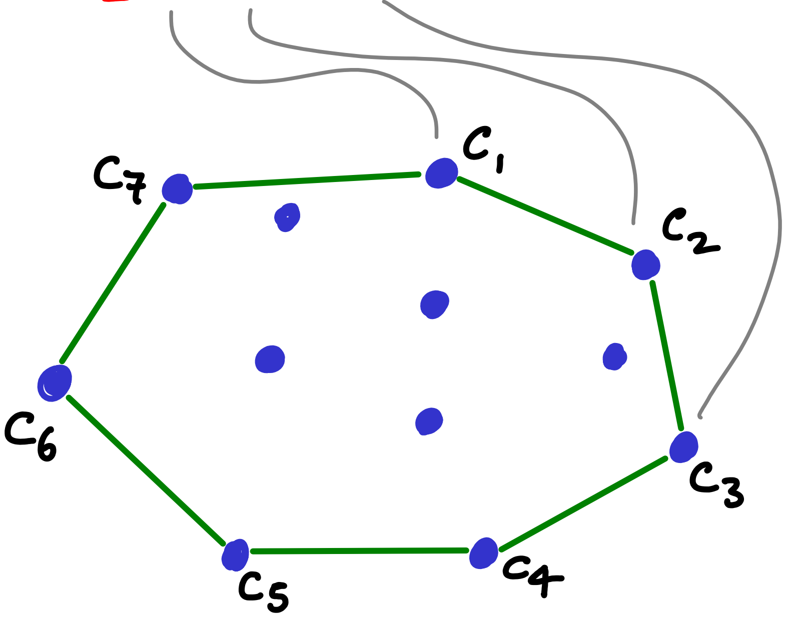
C.H. input: list of (2D) points

$$[P_{10}\ \ P_5\ \ P_3\ \ \underbrace{P_{12}}_{(x_{12},\, y_{12})}\ \ P_1\ \ldots \text{etc}\ldots]$$



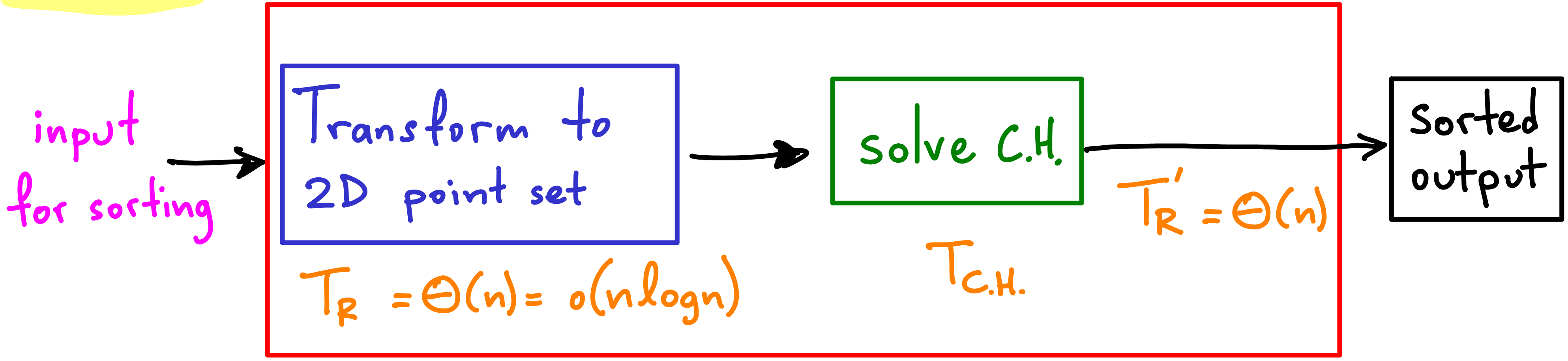
C.H. output: a graph (cycle) containing only the "extreme" points

$$[P_3\ \ P_2\ \ P_{13}\ \ P_5\ \ P_{10}\ \ P_6\ \ P_8]$$

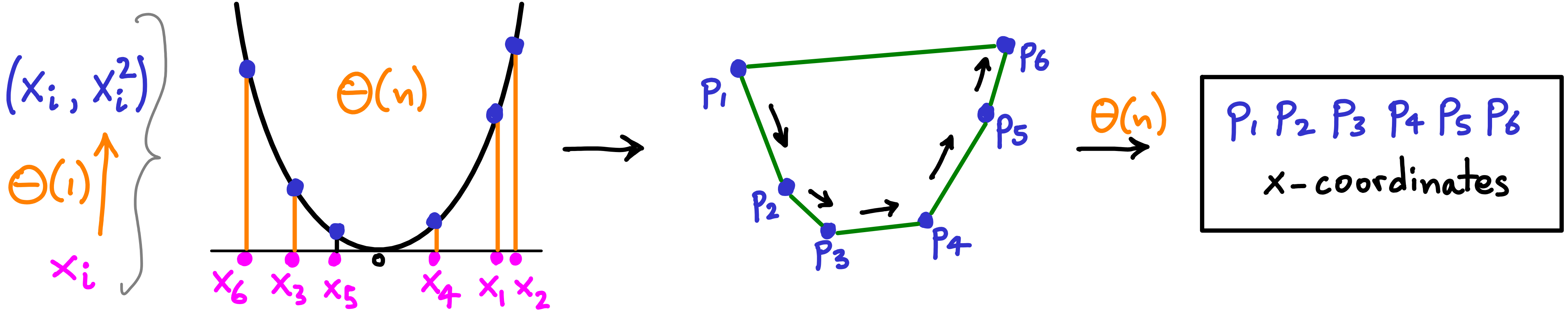


LOWER BOUND FOR ONE ~~(DECISION)~~ PROBLEM VIA ANOTHER

Example



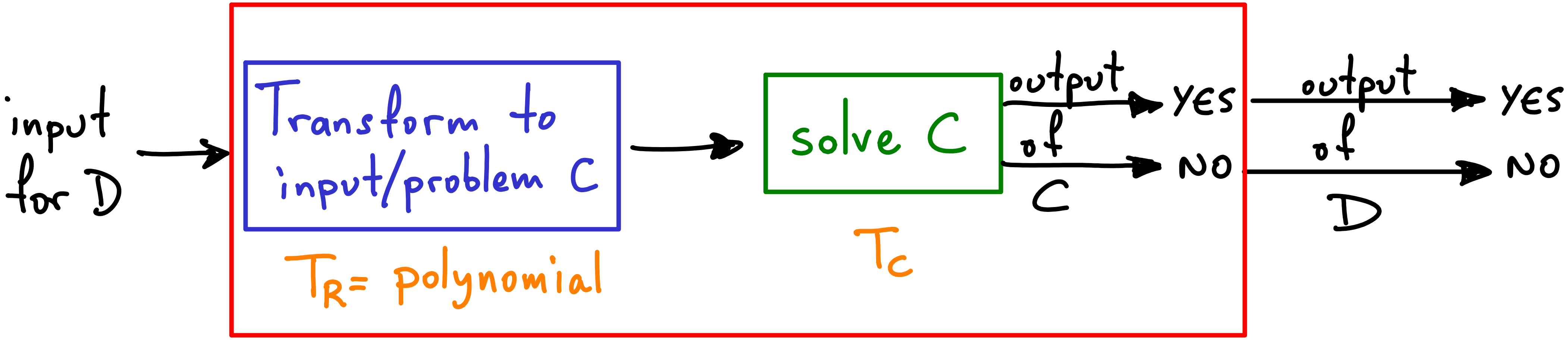
solve sorting: known $\Omega(n\log n) = T_R + T_R' + T_{C.H.} \Rightarrow T_{C.H.} = \Omega(n\log n)$



NPC REDUCTION

FROM ONE (DECISION) PROBLEM D TO ANOTHER C

~LIKE ESTABLISHING AN ALMOST CERTAIN LOWER BOUND FOR C.



solve D: known NPC $= T_R + T_C =$ poly-time $+ T_C$

$\Rightarrow$ C is NP-hard

(if T_C is polynomial $\Rightarrow P = NP$)

& if C is in NP then C is NPC

$A \leq_p B$: in polynomial time, A can be reduced to B

↳ B is at least as hard as A.

Solving B in poly-time $\Rightarrow$ solving A in poly-time

If $\{\text{every problem in NP}\} \leq_p B$ then B is NP-hard

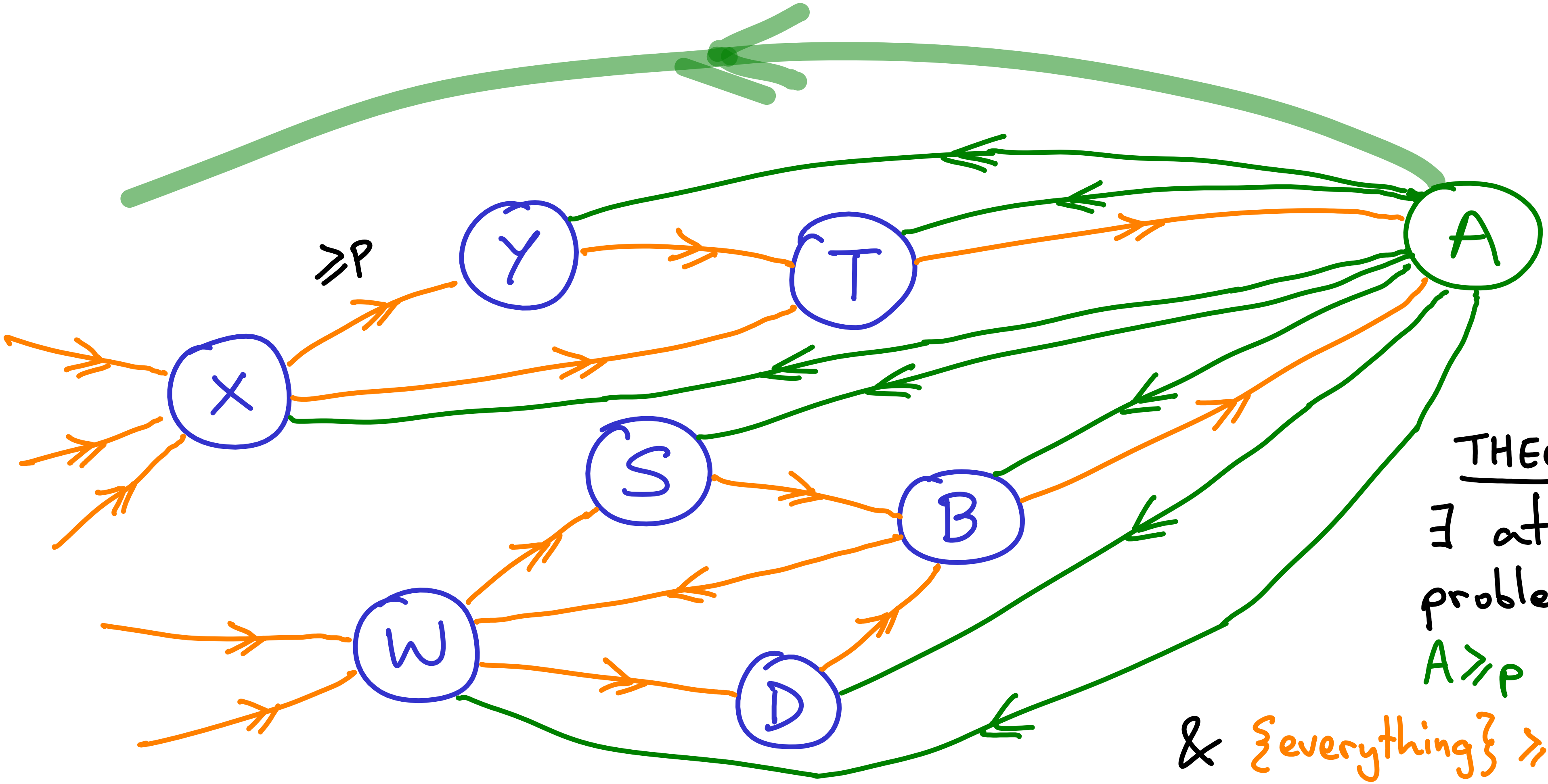
↳ or $\{\text{any NPC problem}\}$

(and if B is in NP, then B is NPC)

if A & B are NPC then $\underbrace{A \leq_p B \quad \& \quad B \leq_p A}_{\sim\ A =_p B}$

THOUSANDS OF NPC PROBLEMS

Reductions/transformations between them resemble a di-graph.



strongly connected components?

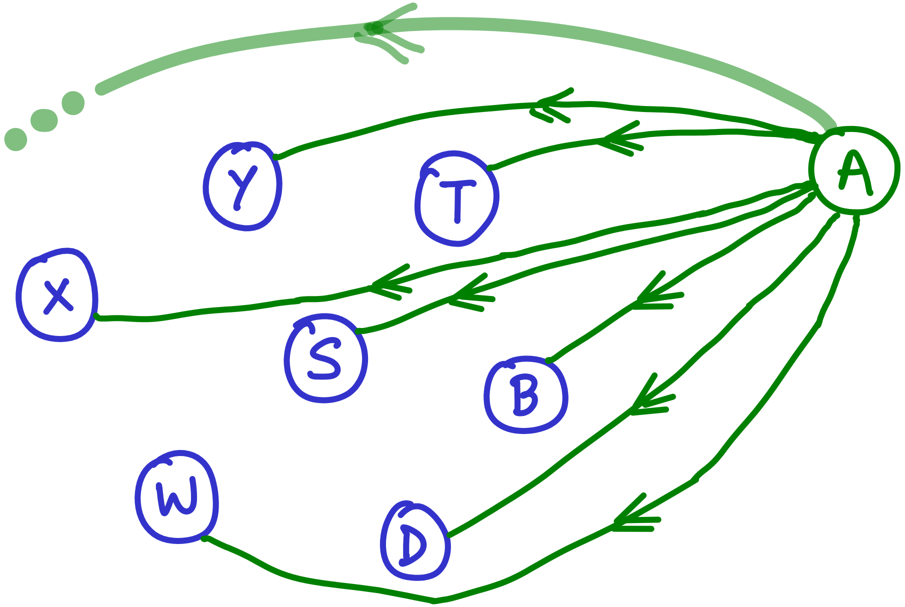
Must be 1. precisely.

THEOREM

∃ at least one problem A s.t.

$A \geq_p \{\text{everything}\}$

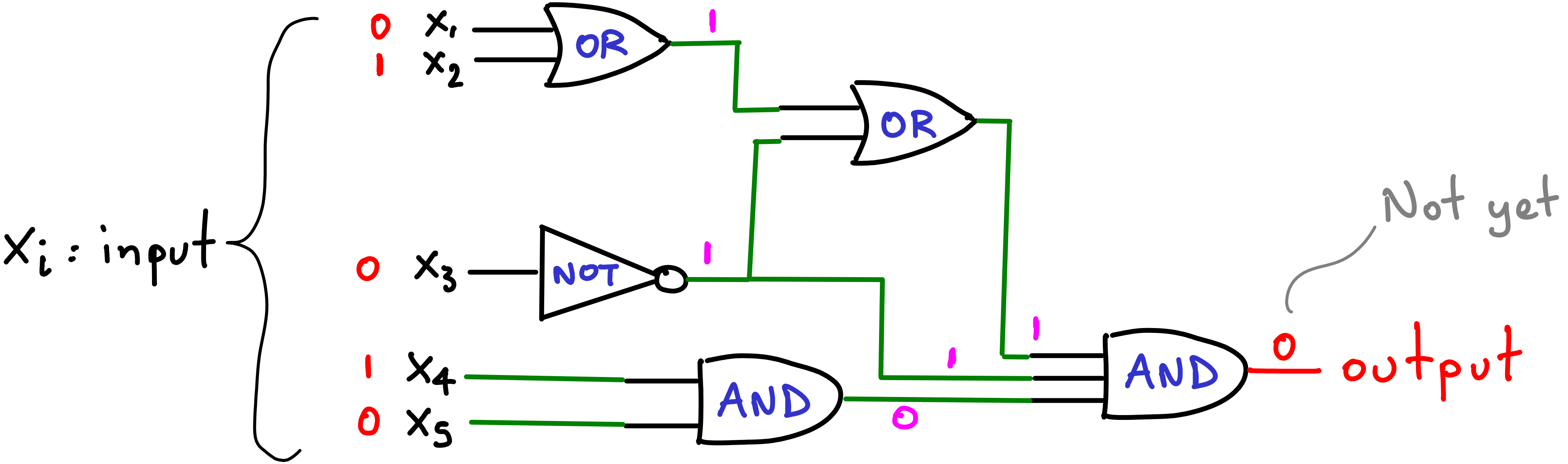
& $\{\text{everything}\} \geq_p \dots \geq_p \dots \geq_p A$

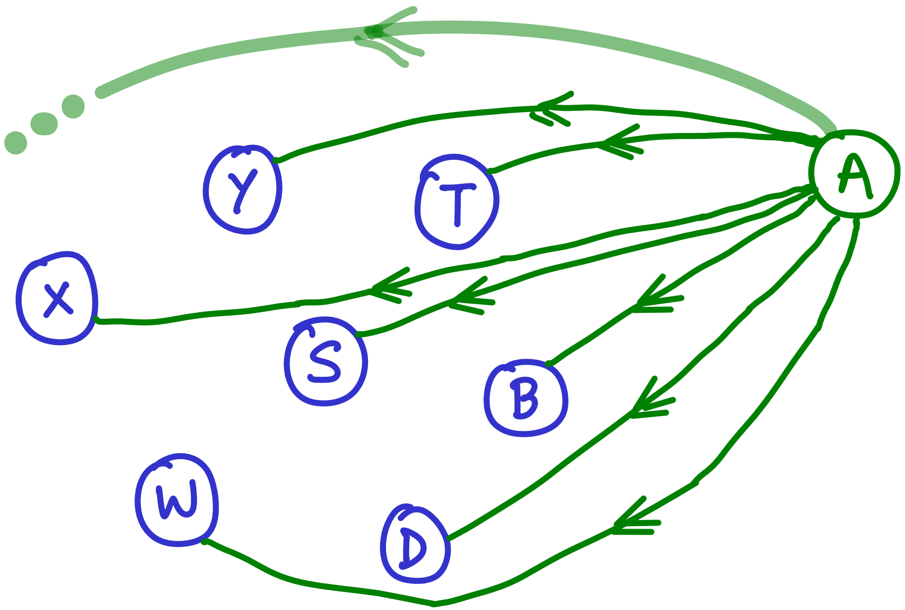


Circuit SAT (satisfiability)

The first NPC problem.

Given a circuit, can the output ever be 1?





Circuit SAT (satisfiability)

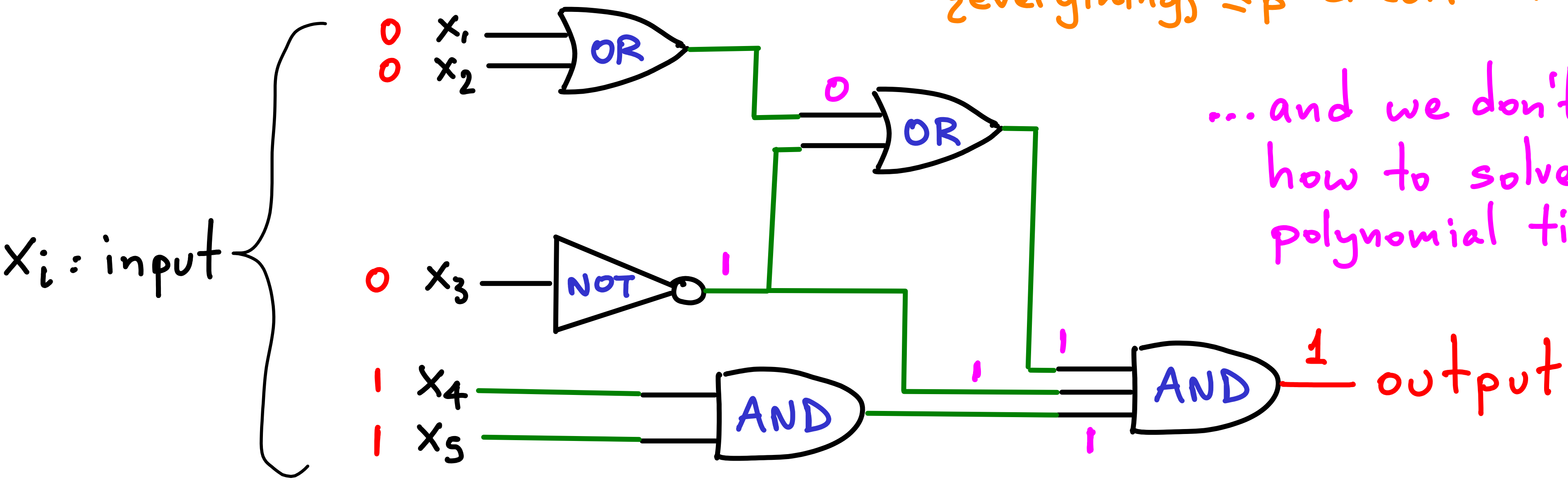
The first NPC problem.

Given a circuit, can the output ever be 1?

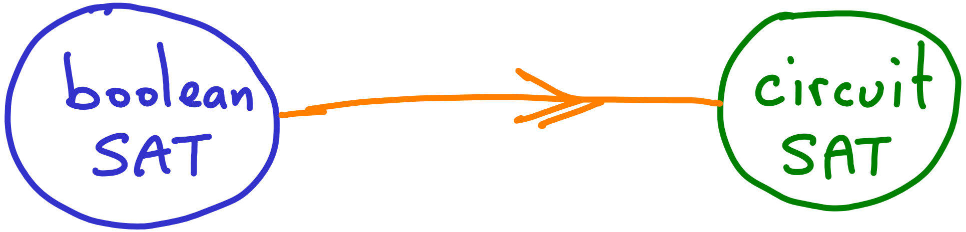
1 – in NP ... intuitive

2 ~ every problem can be described as a circuit — technical

$\{\text{everything}\} \leq_p$ circuit-SAT



...and we don't know how to solve this in polynomial time. (not in P)

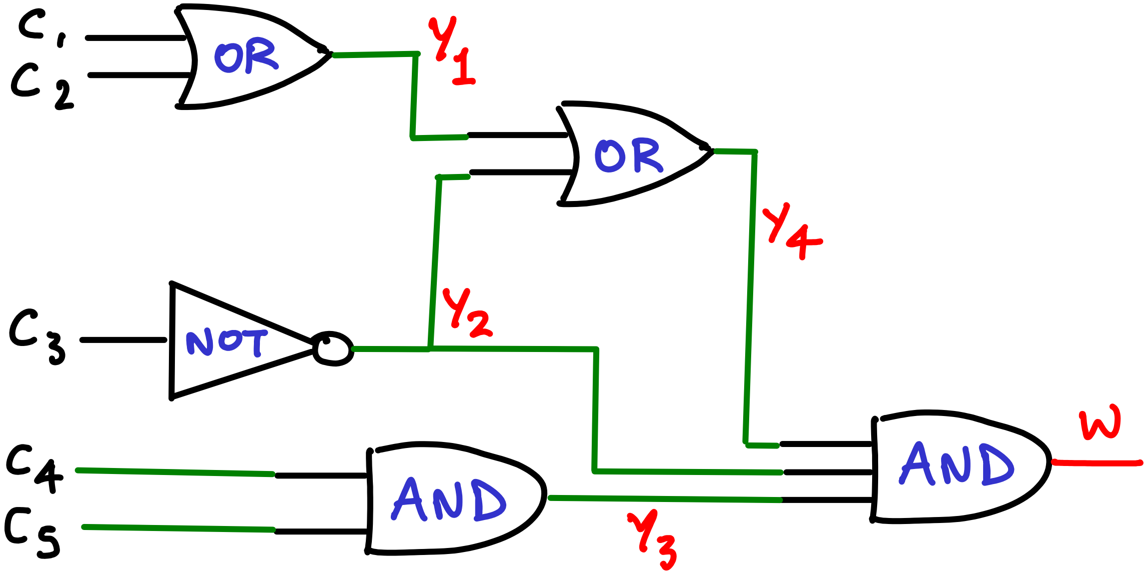


circuit-SAT $\leq_p$ Boolean SAT

(actually also prototypical)

Boolean SAT $\quad (x_1 \vee x_2 \vee \overline{x_3}) \wedge ((x_1 \leftrightarrow x_5) \vee (x_4 \to \overline{x}_3)) \;\to ? \to\; 1$

Given a circuit-SAT instance transform (quickly) into a Boolean SAT_



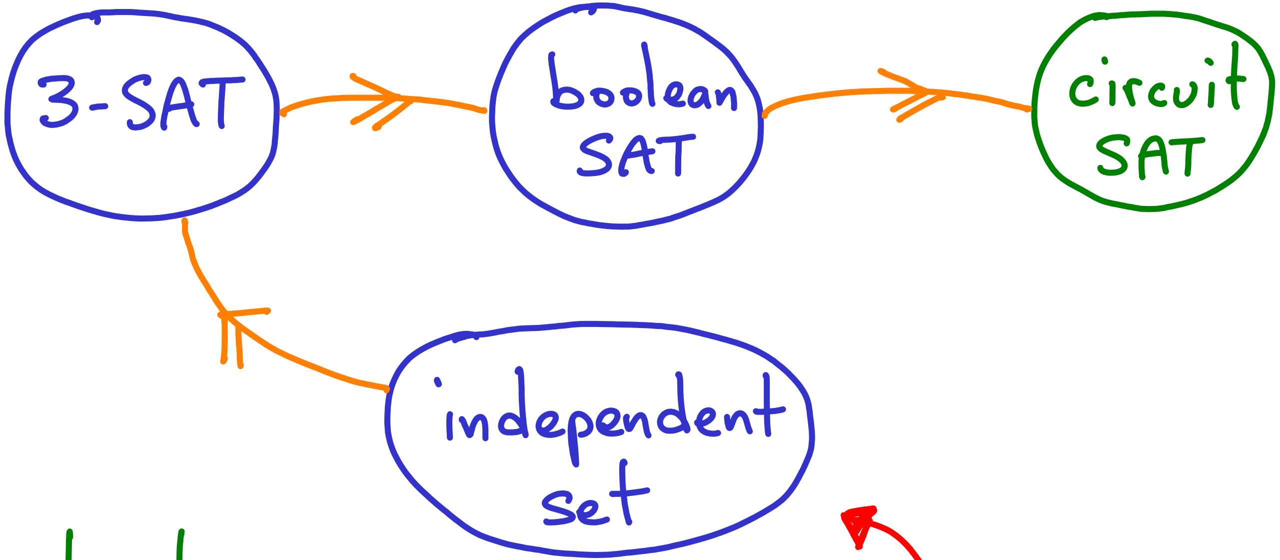
$$
\begin{aligned}
& w \wedge (w \leftrightarrow (y_4 \wedge y_2 \wedge y_3)) \\
& \wedge (y_4 \leftrightarrow (y_1 \vee y_2)) \\
& \wedge (y_1 \leftrightarrow (c_1 \vee c_2)) \\
& \wedge (y_2 \leftrightarrow \overline{c_3}) \\
& \wedge (y_3 \leftrightarrow (c_4 \wedge c_5))
\end{aligned}
$$

3-SAT $\rightarrow$ boolean SAT $\rightarrow$ circuit SAT

3-SAT

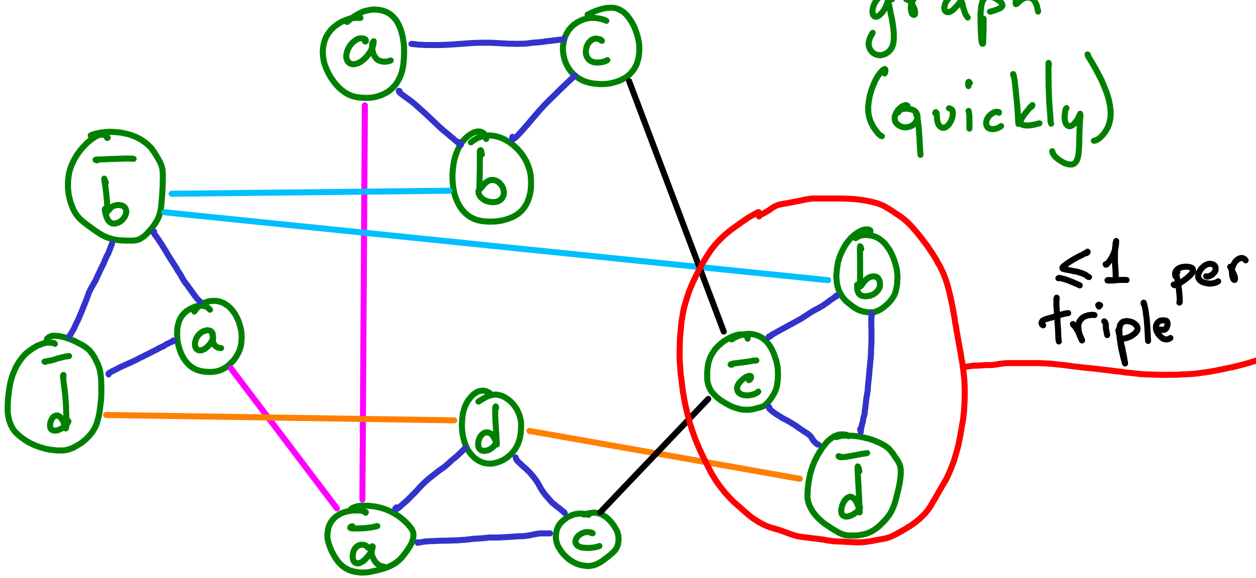
$$\left.\begin{array}{l} \quad (x_1 \vee x_2 \vee x_3) \\ \wedge\ (x_1 \vee x_4 \vee x_5) \\ \wedge\ (x_2 \vee \overline{x}_4 \vee x_{13}) \\ \wedge\ (x_2 \vee \overline{x}_3 \vee x_3) \\ \qquad\quad \vdots \end{array}\right\} \ k \text{ clauses}$$

each w/ 3 literals



$$\left.\begin{aligned}&(a\vee b\vee c)\\ \wedge\,&(b\vee\bar c\vee\bar d)\\ \wedge\,&(\bar a\vee c\vee d)\\ \wedge\,&(a\vee\bar b\vee\bar d)\end{aligned}\right\}\ k \text{ clauses}$$

construct graph (quickly)

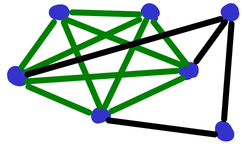


Ask: does this graph have k independent vertices?

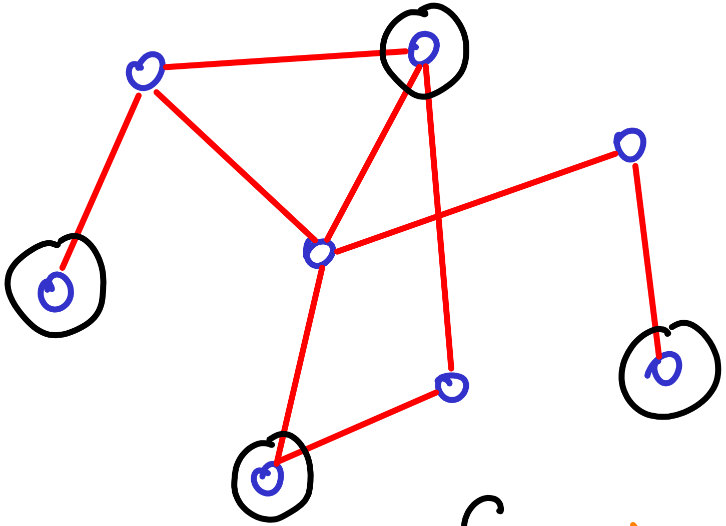
If yes, then 3-SAT is ✓

If no, then 3-SAT is ✗

CLIQUE in a graph: subset of V s.t. all pairs of vertices share edges.

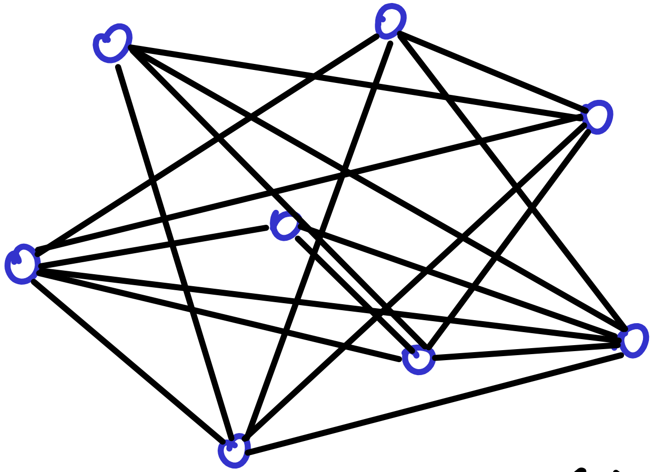


size k independent set in G? $\iff$ size k clique in G^c?

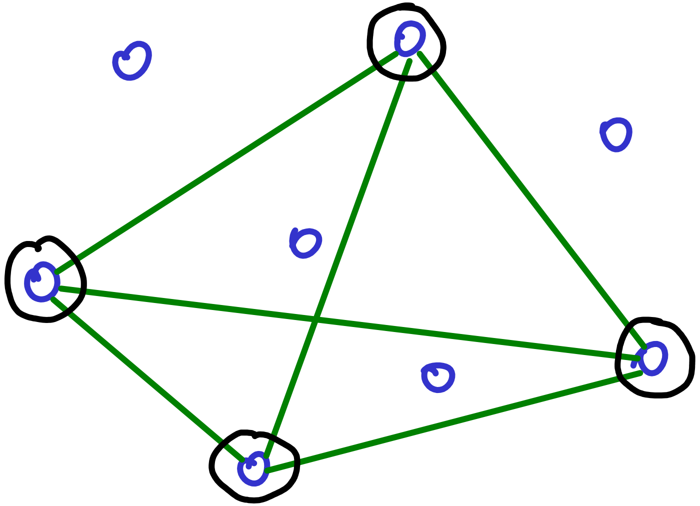


G

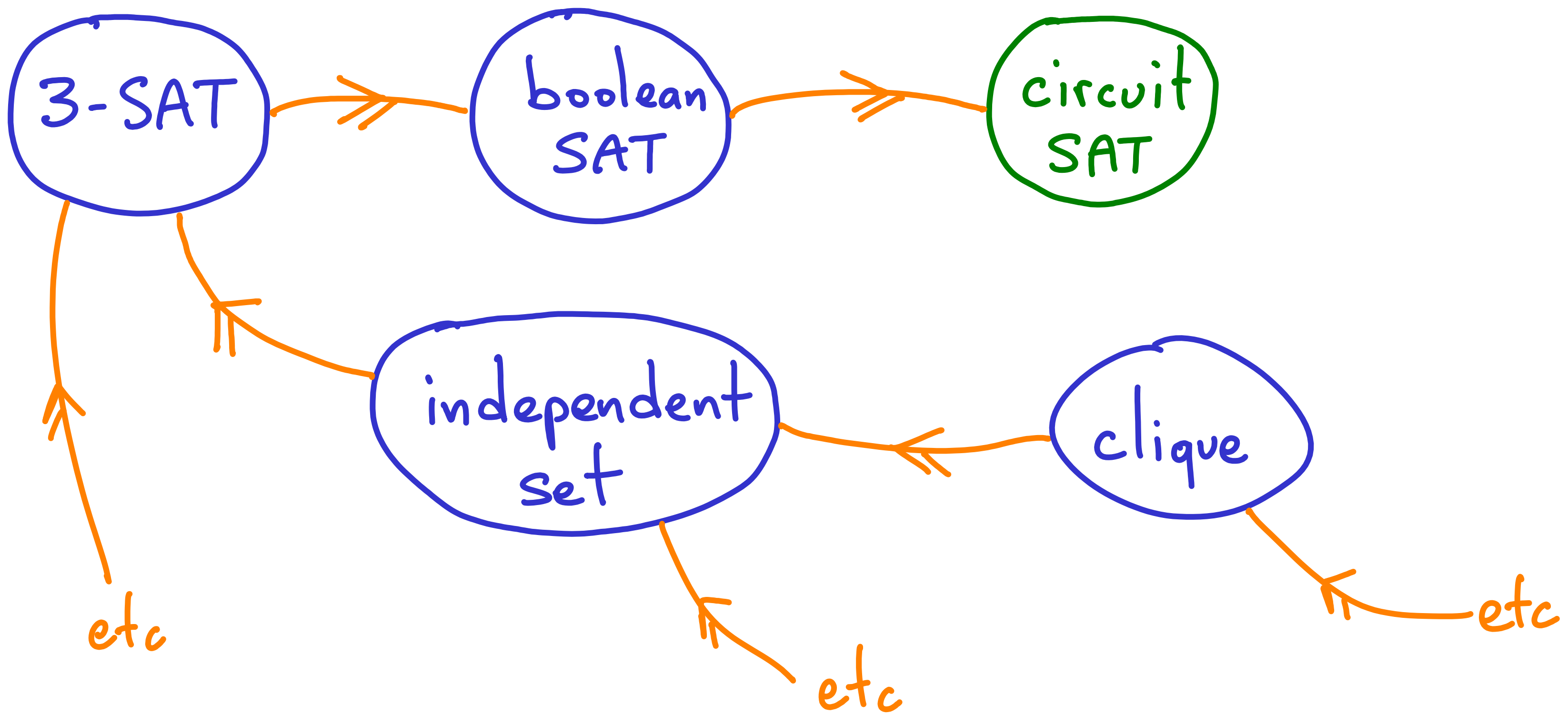
transform



complement (G) = G^c



Asking if a graph has a clique of size k : NPC



3-SAT
boolean SAT
circuit SAT
independent set
clique
etc
etc
etc

OTHER HARD PROBLEMS

- knapsack: given item types, w/ size & value, fill a bag w/ max value (multiples OK)
- subset sum: does a subset of given integers sum to t?
- partition: split set of integers in 2 groups with equal sums
- planar SAT: SAT without wire crossings
- set cover: given some sets, select min# sets s.t all elements are present
- hitting set: given sets, select min# elements s.t. all sets are represented
- longest path: visiting each vertex once.
- Steiner tree: ~MST on selected subset of a graph
- traveling salesman: min-cost simple cycle on weighted complete graph

OTHER HARD PROBLEMS

... & some are even harder

Tetris

Minesweeper

Lemmings

Mario Bros

Pac-man

Prince of Persia

Portal

Doom

etc