L26

goo.gl/w3LvmN

4102

4.26.2016

abhi shelat

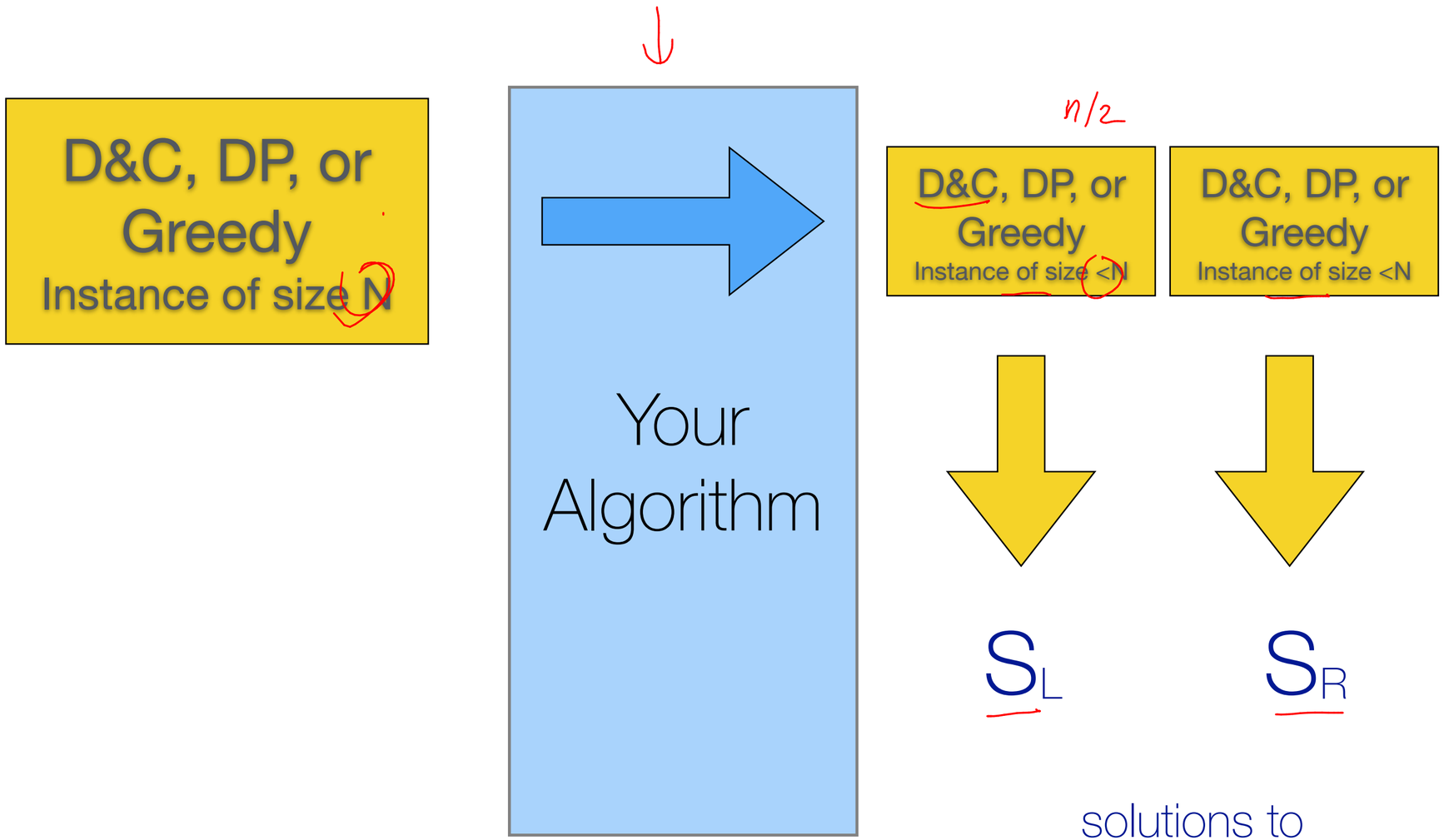
WE HAVE BEEN SOLVING PROBLEM A BY SOLVING SMALLER VERSIONS OF PROBLEM A

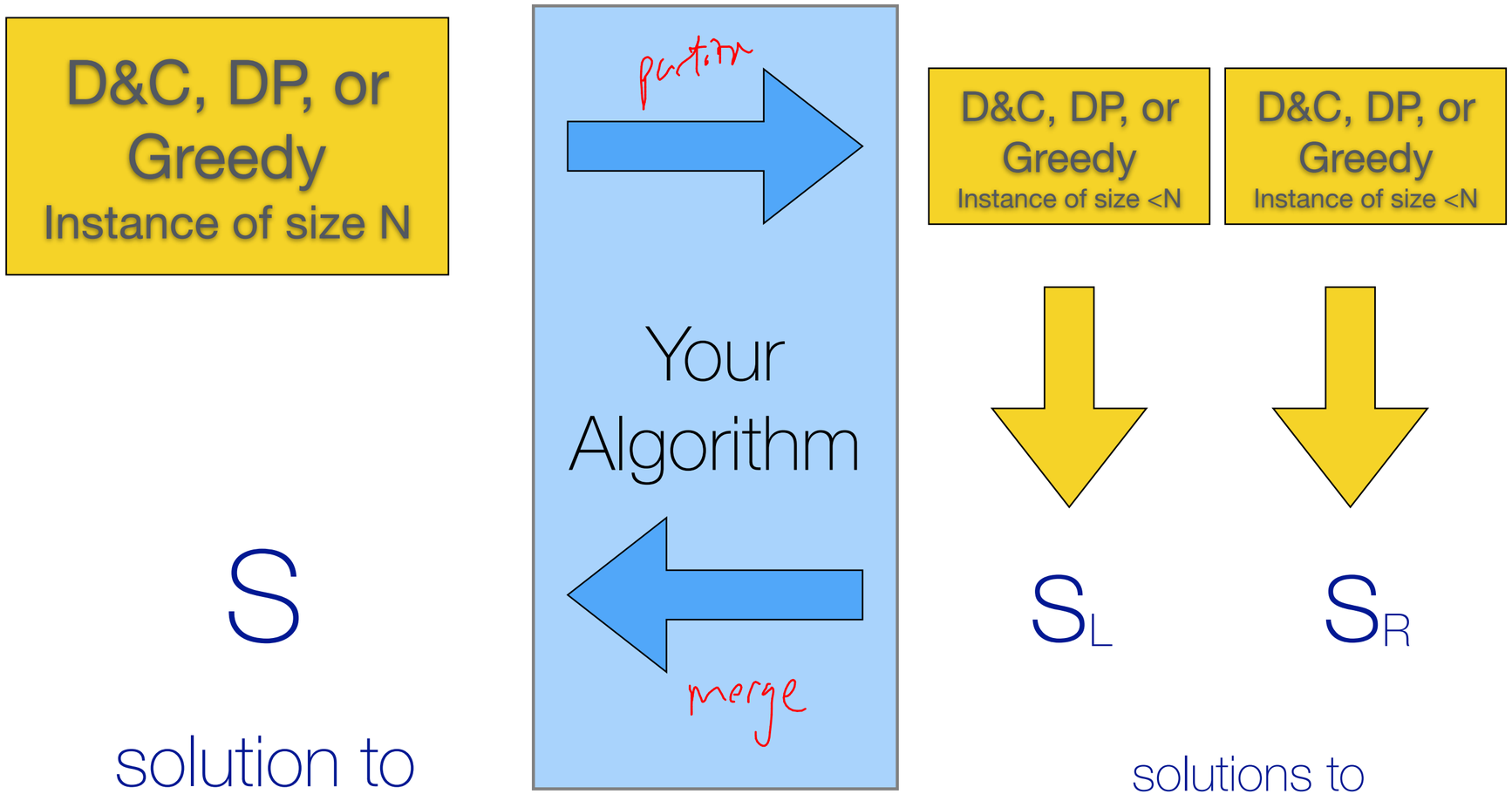
GENERAL IDEA:

SOLVE PROBLEM A BY SOLVING PROBLEM B

D&C, DP, or Greedy

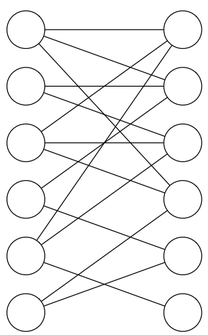
Instance of size N


D&C, DP, or Greedy
Instance of size N
Your Algorithm
D&C, DP, or Greedy
Instance of size <N
D&C, DP, or Greedy
Instance of size <N
n/2
S_L
S_R
solutions to smaller instance

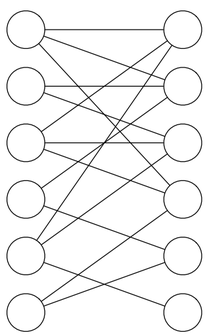

D&C, DP, or Greedy
Instance of size N
partition
Your Algorithm
merge
D&C, DP, or Greedy
Instance of size <N
D&C, DP, or Greedy
Instance of size <N
S_L
S_R
S
solution to original problem
solutions to smaller instance

(4R4E)

Bipartite Matching Instance

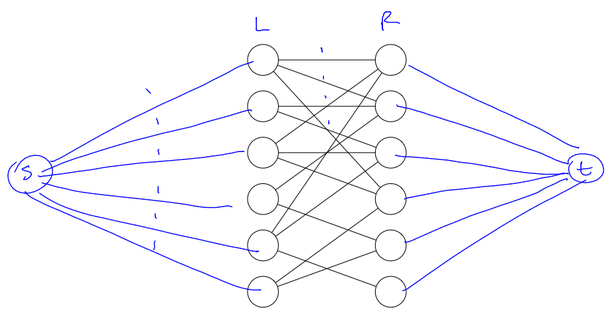


Bipartite Matching Instance

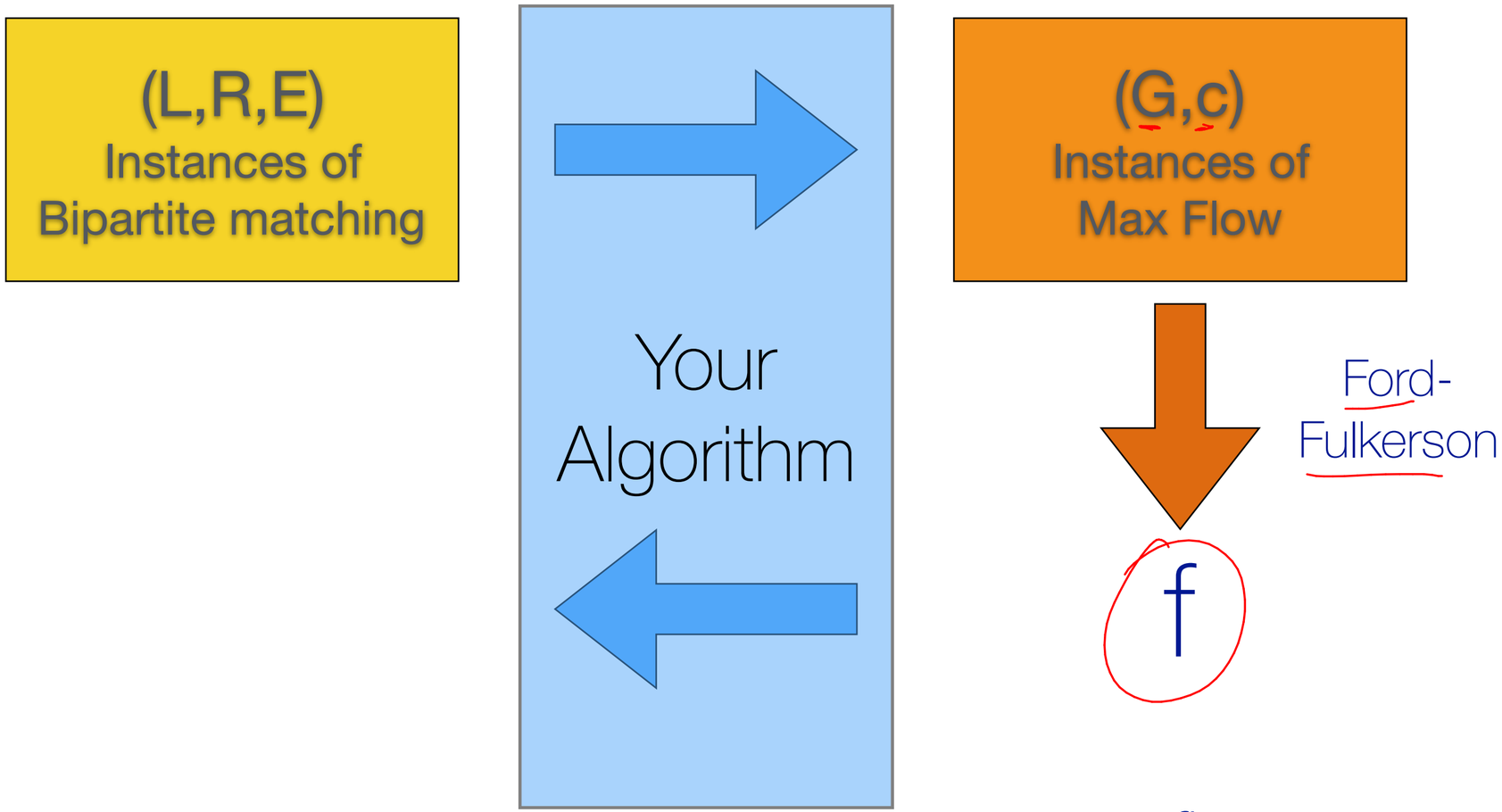


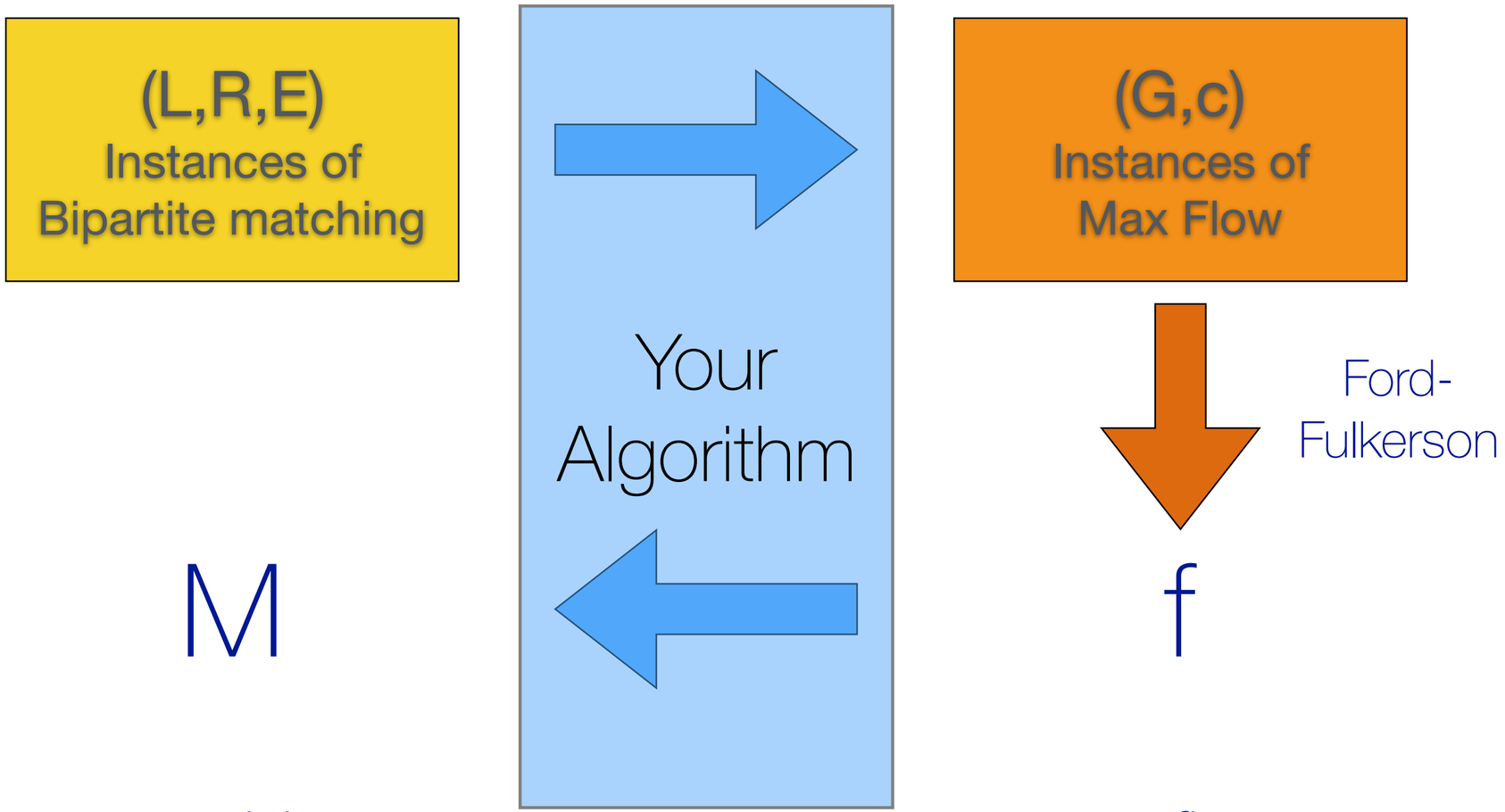
Max flow Instance

(G, c)



(L,R,E)
Instances of
Bipartite matching

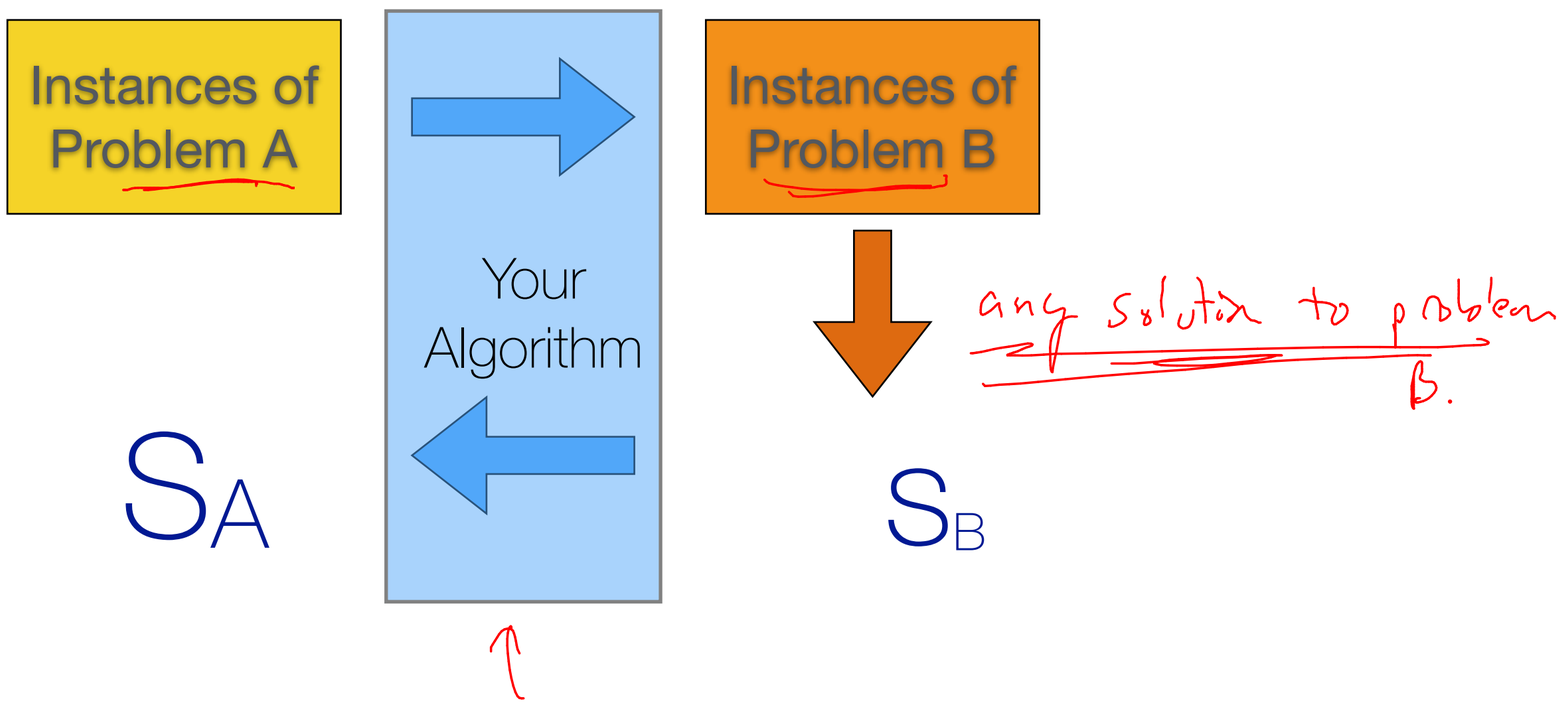

(L,R,E)
Instances of
Bipartite matching
Your
Algorithm
(G,c)
Instances of
Max Flow
Ford-
Fulkerson
f
flow


(L,R,E)
Instances of
Bipartite matching
Your
Algorithm
(G,c)
Instances of
Max Flow
Ford-
Fulkerson
M
f
matching
flow

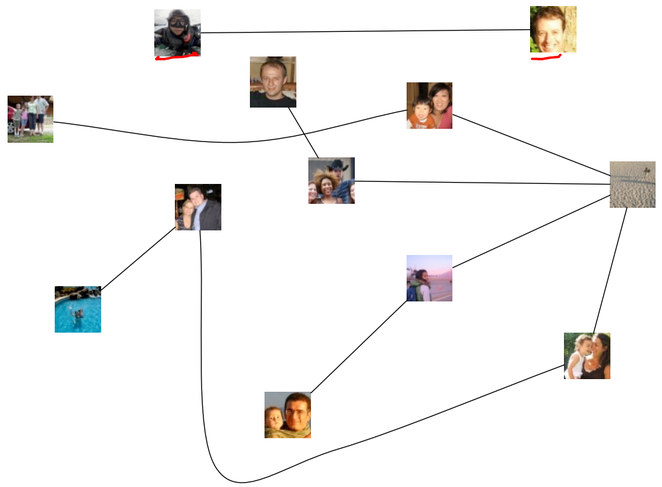
Reduction

"as hard as solving problem A."

$$\text{PROBLEM}_a \leq_{f(n)} \text{PROBLEM}_b$$



party problem



draw an edge b/w people who do not get along.

largest

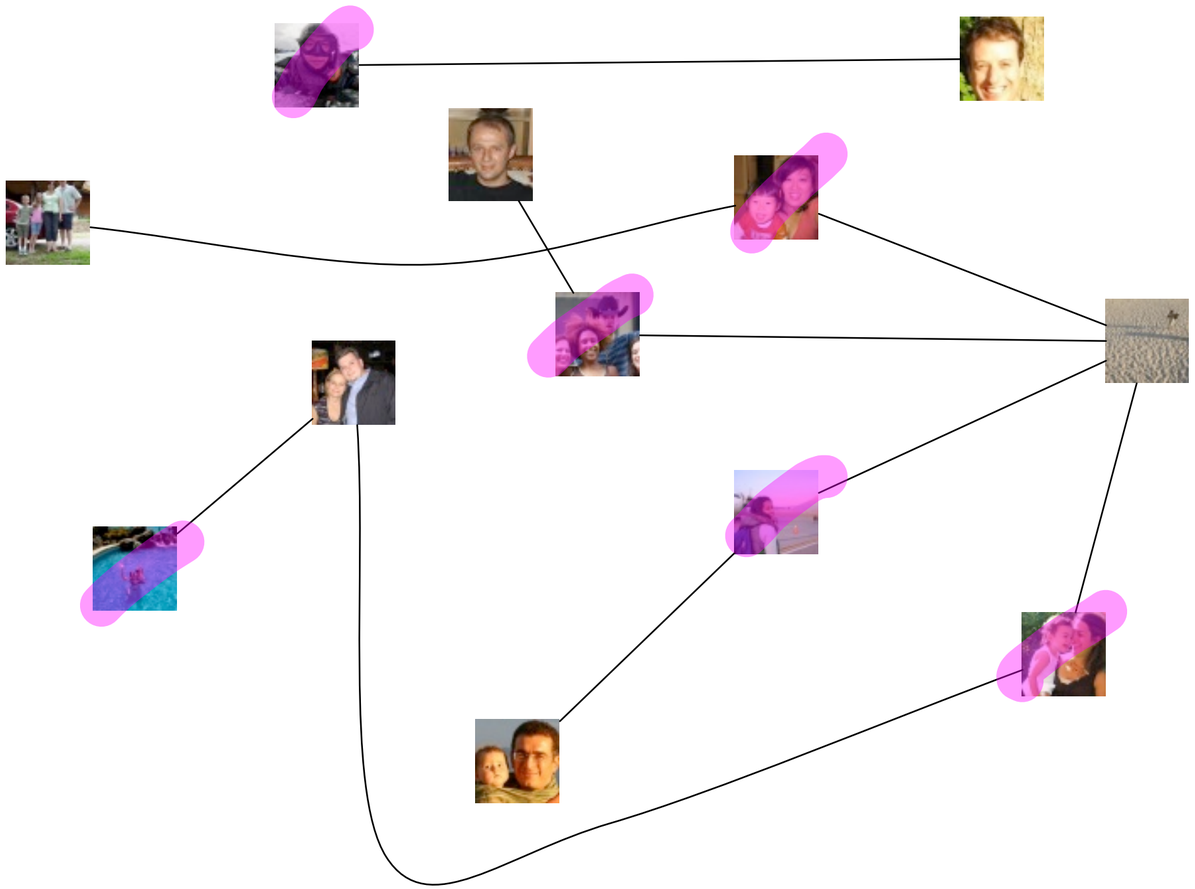
find a set of people who "get along" in this graph

independent set

a set $S \subseteq V$ is an independent set if

no two nodes in S are joined by an edge.

example

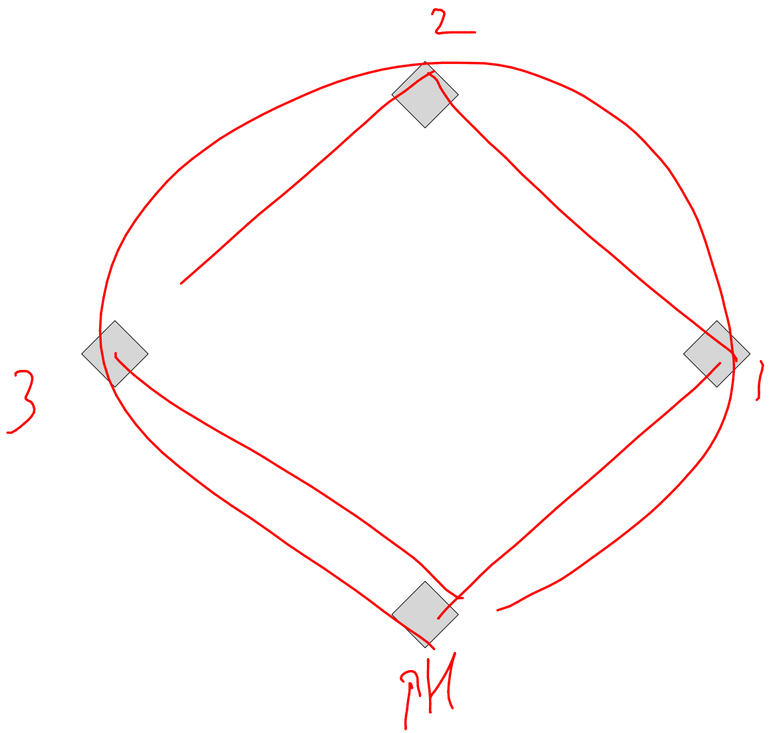


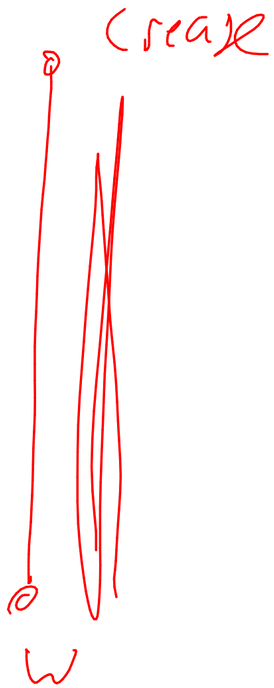
- No two nodes joined by an edge.

goal:

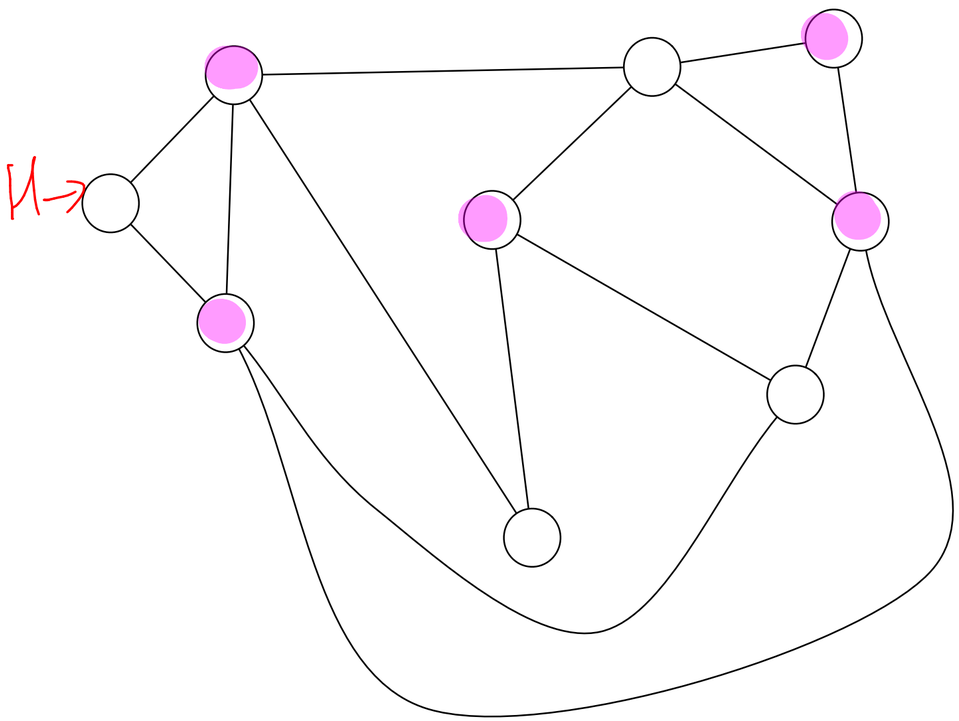
Given a graph G, find the MAX INDEPENDENT SET.

baseball





baseball



how to defend such a graph ??

→ want some defender on every edge

Smallest such set

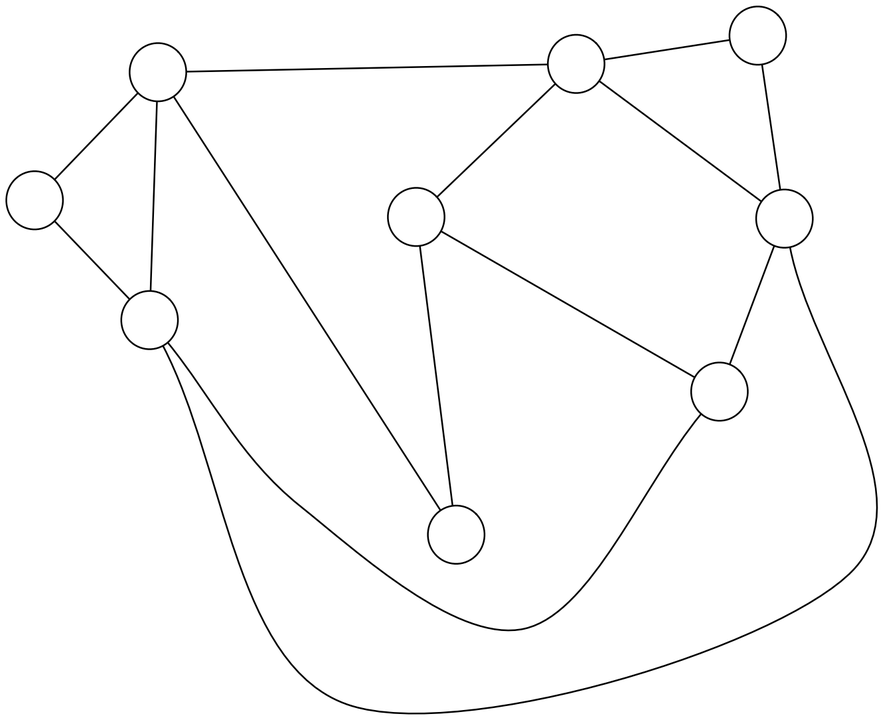
A vertex cover of a graph is a subset $S \subseteq V$ such that for each edge $e = (x, y) \in E$, either $x \in S$ or $y \in S$.

a vertex cover of a graph is a set $C \subseteq V$

such that $\forall\ (x, y) \in E$

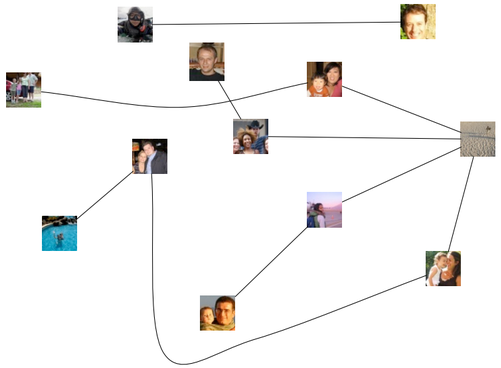
either $x \in C$ or $y \in C$

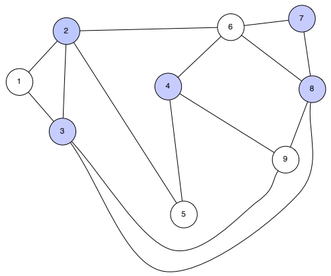
example



goal:

given a graph G, FIND the MINIMUM SIZE vertex cover for G.

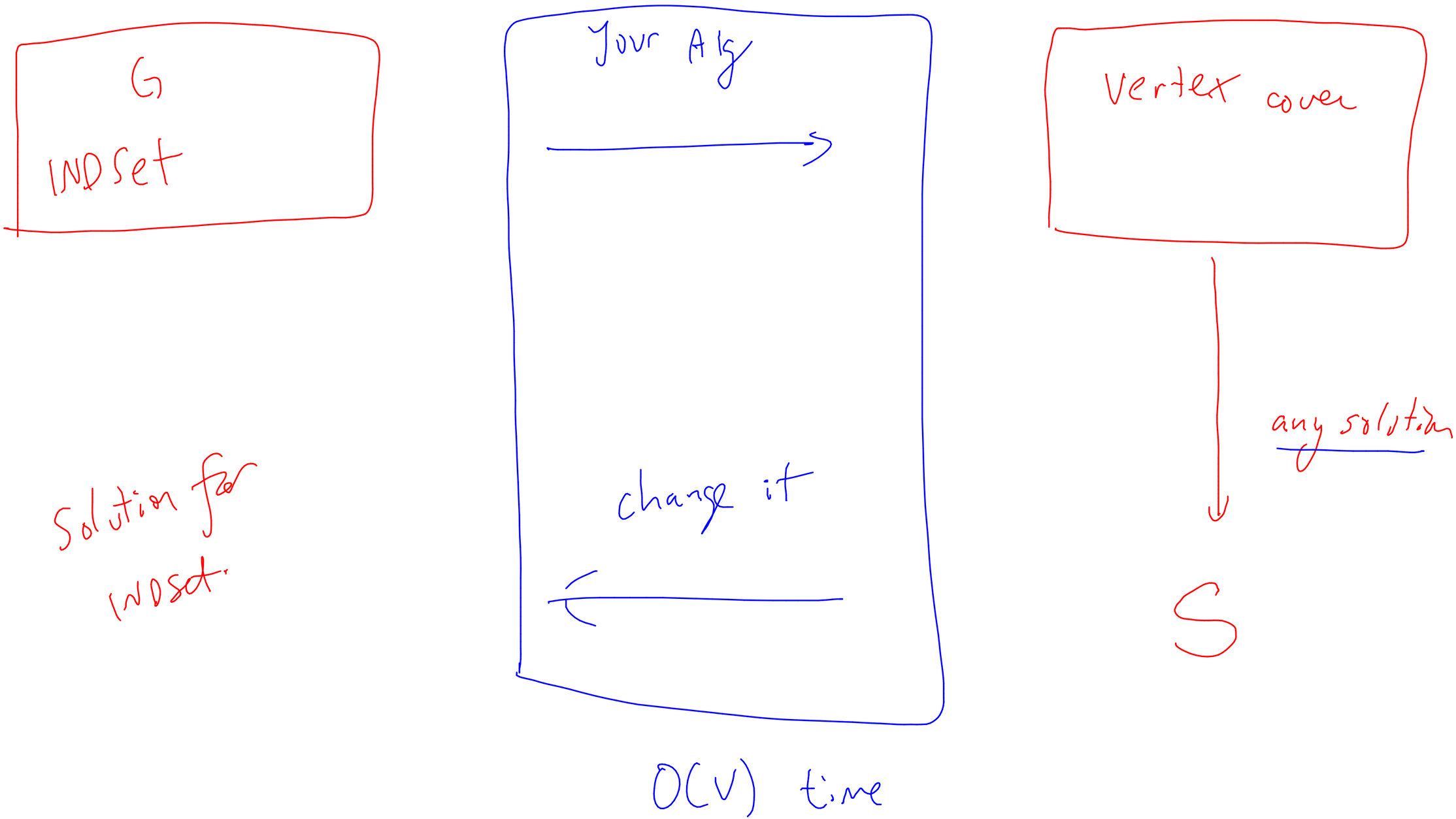




① VC problem is "AS HARD AS" the INDSET problem.

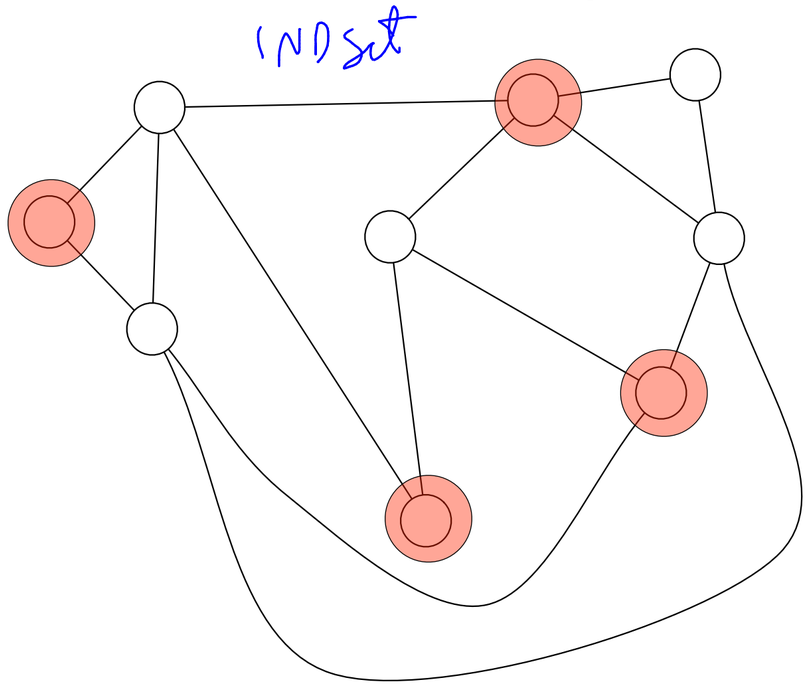
$$\text{MAXINDSET} \leq_{O(V)} \text{MINVERTEXCOVER}$$

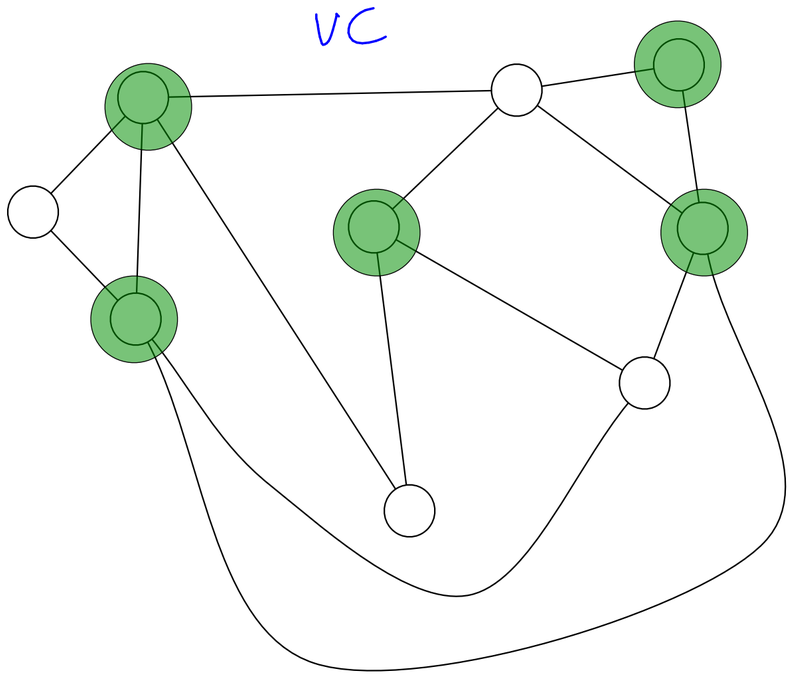
A solution to VC can be used to solve INDSET.


G
INDSet
Your Alg
change it
Vertex cover
any solution
S
Solution for INDSet.
O(V) time

Thm: set S is VC of graph G $\iff$ $V-S$ is an INset of G.

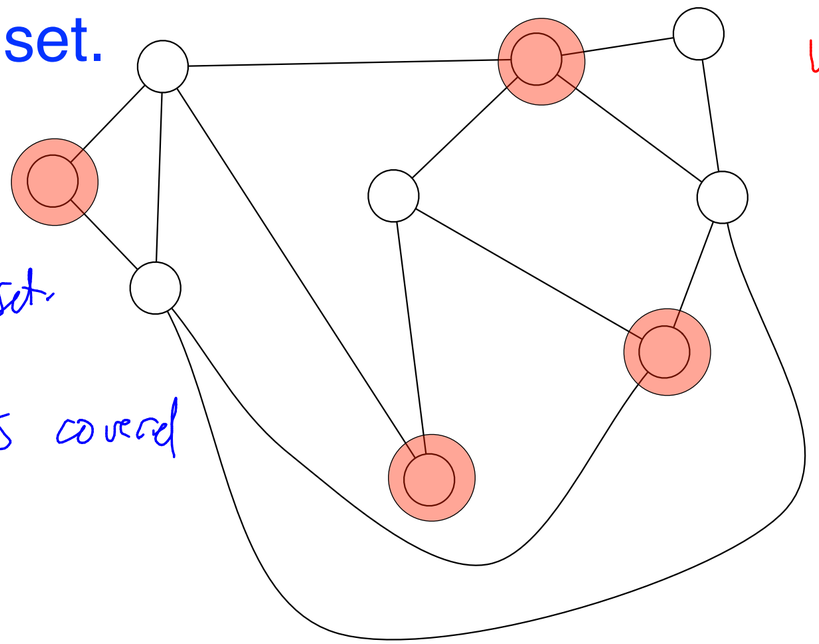
Thm: set S is an independent set of G iff V-S is a vertex cover.





Thm: set S is an independent set of G iff V-S is a vertex cover.

- suppose S is an independent set.



we want to show that V-S is a vertex cover.

Consider any edge $e=(x,y)\in E$.

① $x\in S \Rightarrow y\notin S$ b/c S is indpset.

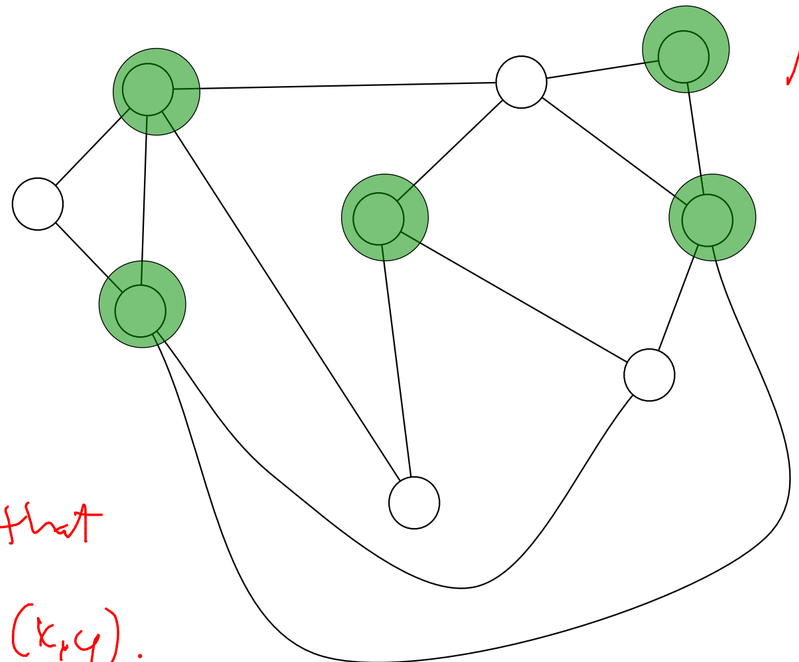
$\Rightarrow y\in V-S$, edge e is covered

② $x\notin S \Rightarrow x\in V-S$, edge e is covered.

$\Rightarrow$ V-S covers every edge in G and so V-S is A VC.

Thm: set S is an independent set of G iff V-S is a vertex cover.

suppose V-S is a vc.



Need to show that S is an IND Set.

Consider an $x \in S$, and

consider any edge $(x,y) \in E$.

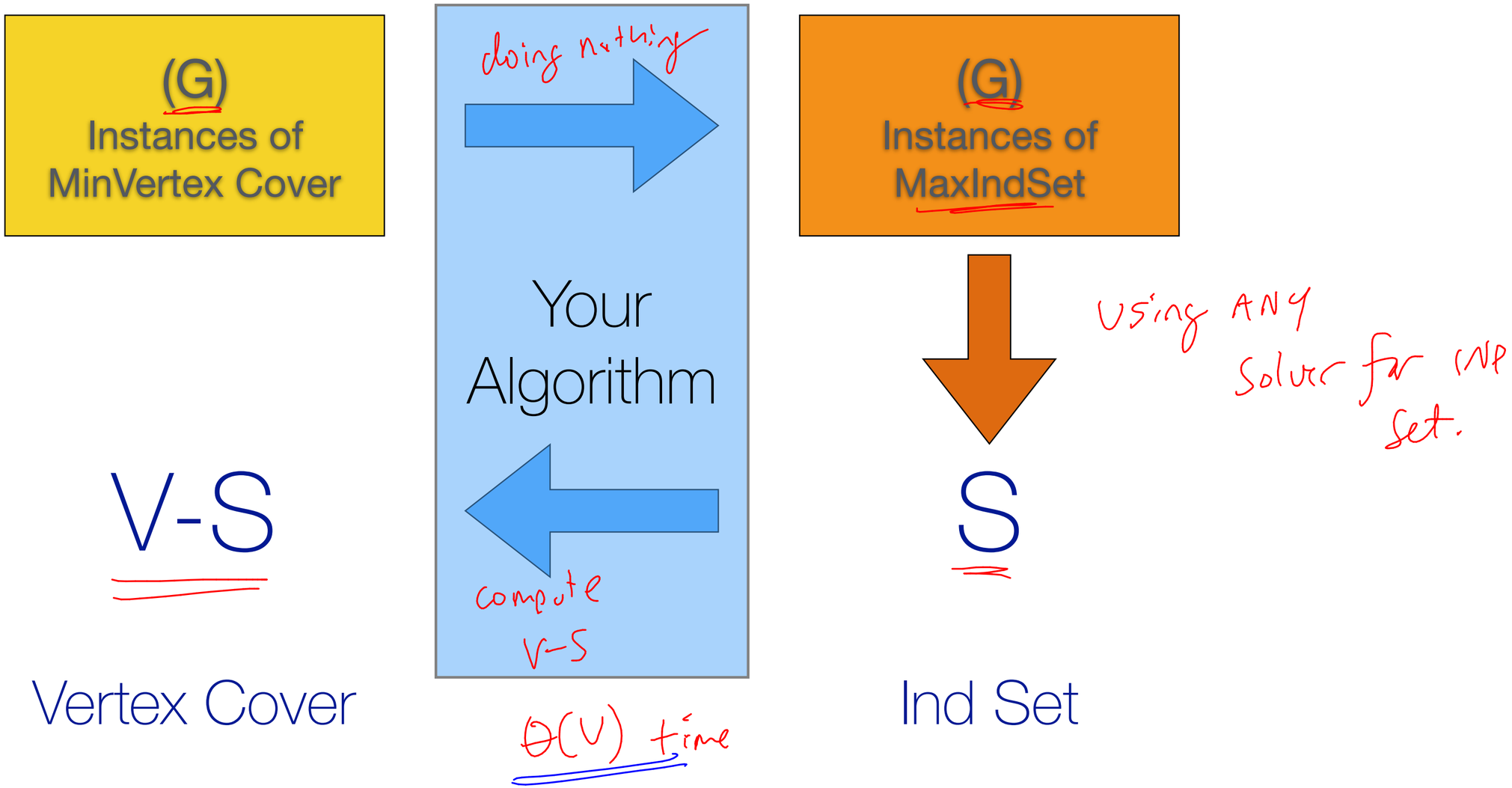
$\Rightarrow$ since $x \notin V-S$, but $V-S$ is a vertex cover, this implies that $y \in V-S$ to cover edge (x,y).

$\Rightarrow y \notin S$. This holds for any $v \in S$ and any edge $(v,w) \in E$.

$\Rightarrow S$ must be an ind. set.

VC ≤ INDSET

INDSET ≤ VC



3sat problem

input: formula in 3CNF form: Logical AND of clauses of 3-variables ORs of

$(a \text{ or } b \text{ or } c)$ AND $(\bar{a} \text{ or } d \text{ or } f)$ AND (---)

clauses; variables

output: "Assignment $A: V \to$ True or false s.t. the formula is TRUE.

3sat example

$(x \vee y \vee z) \wedge (x \vee \bar{y} \vee y) \wedge (u \vee y \vee \bar{z}) \wedge (z \vee \bar{x} \vee u) \wedge (\bar{x} \vee \bar{y} \vee \bar{z})$ = ϕ 5 clause formula

$A(x) \rightarrow T$

$(y) \rightarrow T$

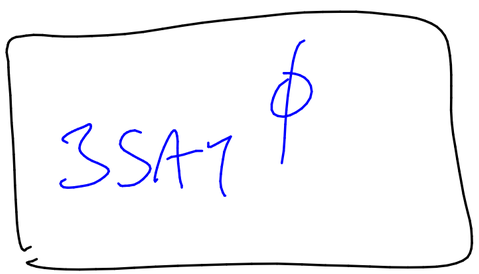
$(u) \rightarrow T$

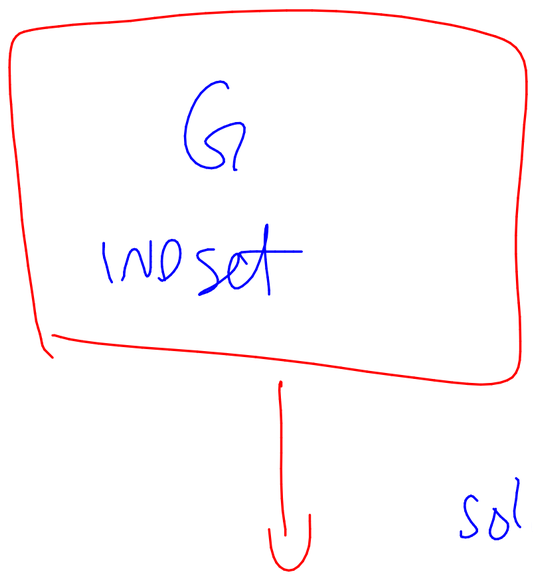
$(z) \rightarrow F$

$3\text{SAT} \leq_p \text{INDSET}$

$$(x \vee y \vee z) \wedge (x \vee \bar{y} \vee y) \wedge (u \vee y \vee \bar{z}) \wedge (z \vee \bar{x} \vee u) \wedge (\bar{x} \vee \bar{y} \vee \bar{z})$$

what must we do to?

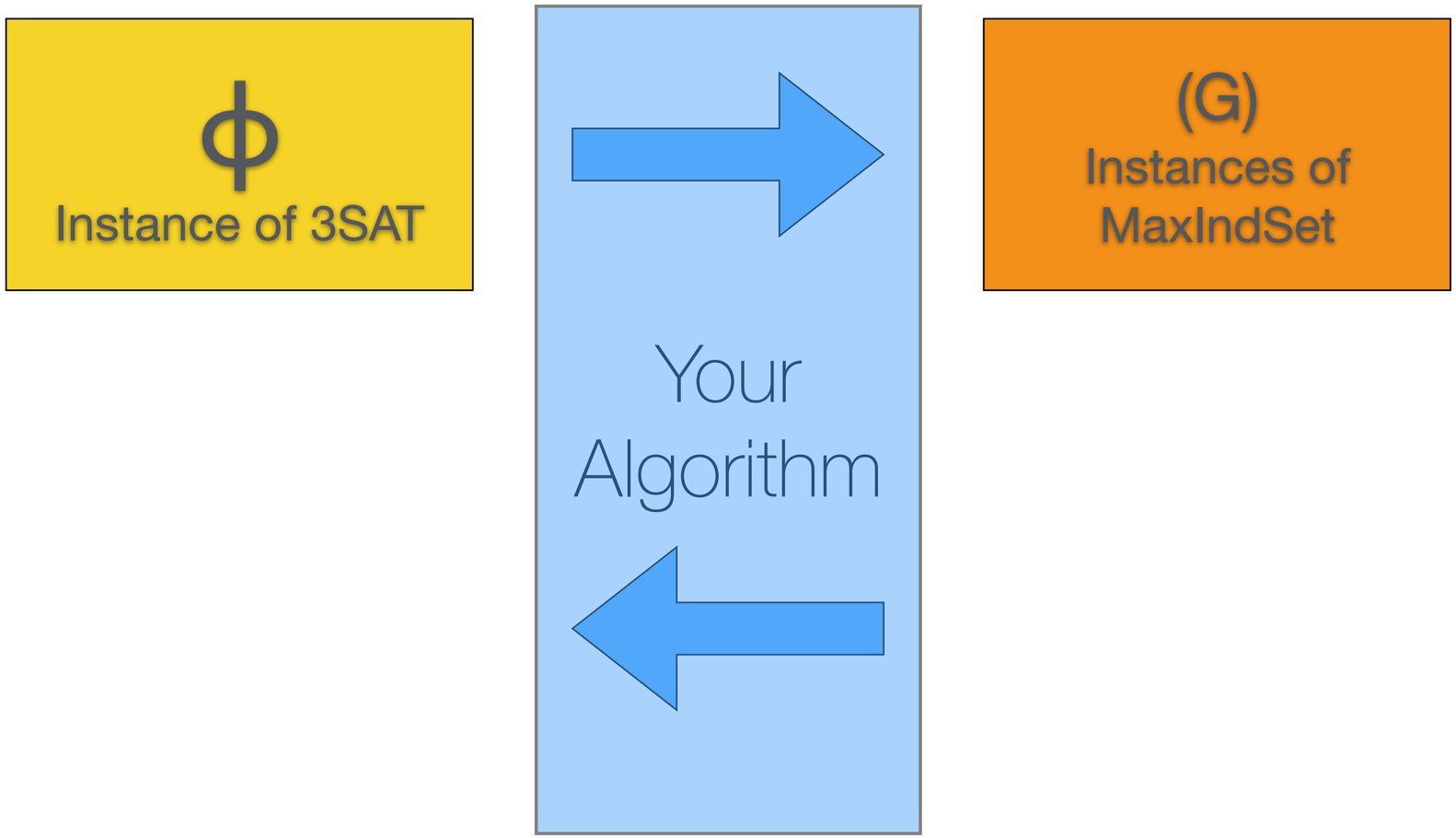


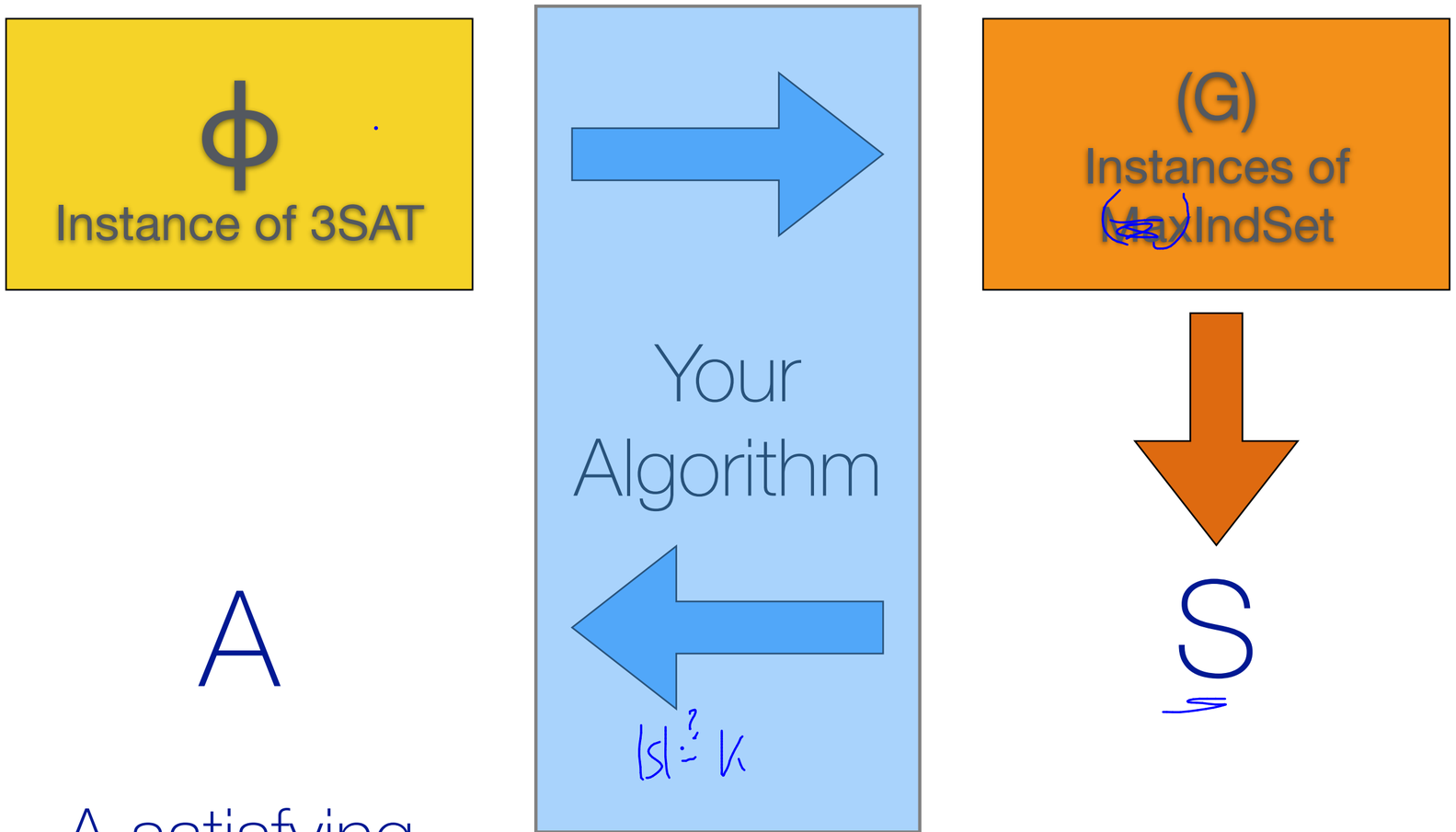


soln

S

Assignment A

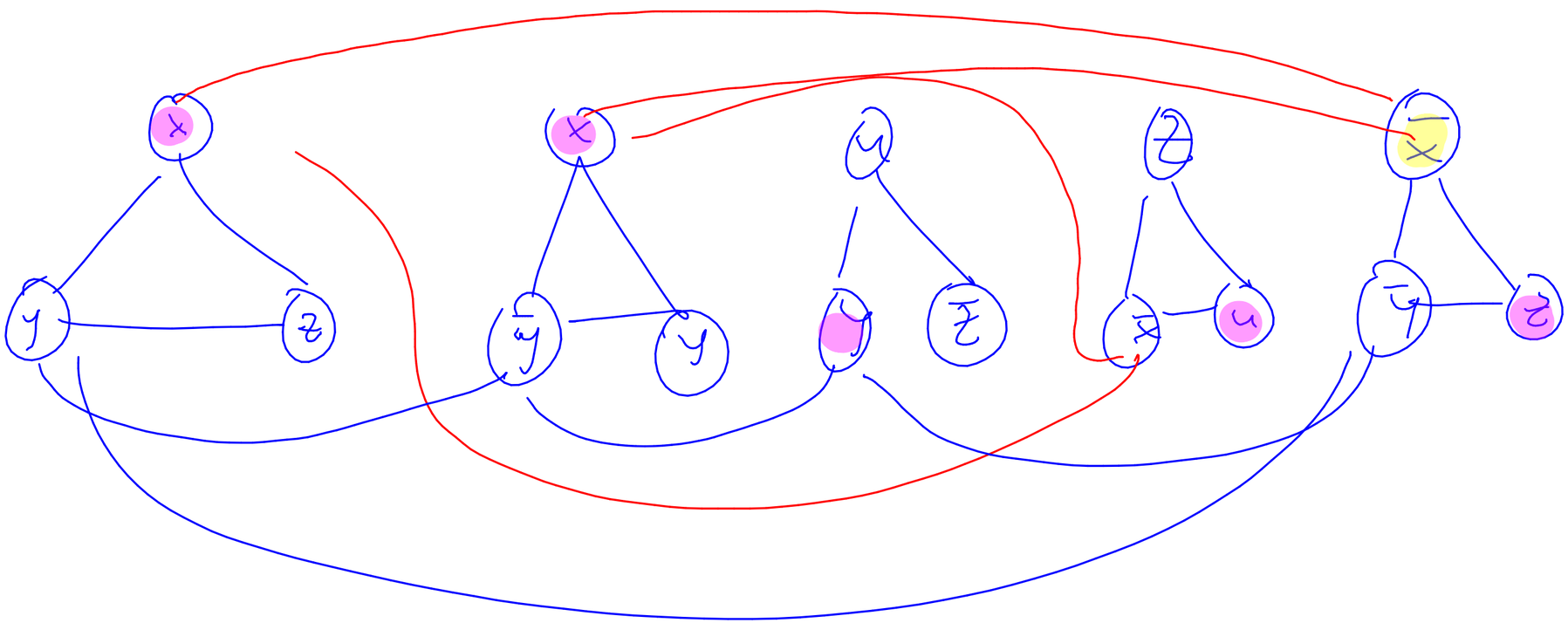

ϕ
Instance of 3SAT
Your
Algorithm
(G)
Instances of
MaxIndSet


ϕ
Instance of 3SAT
Your Algorithm
(G)
Instances of MaxIndSet
S
A
A satisfying assignment
Ind Set

gadgets

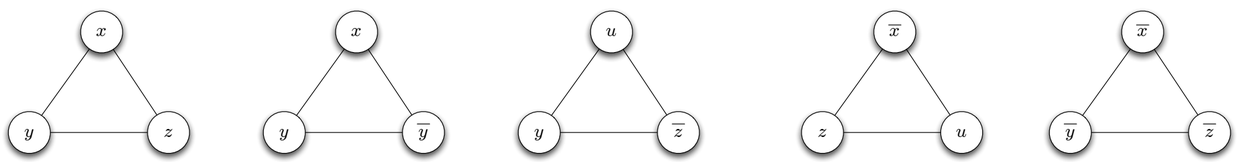
$3\text{SAT} \leq_p \text{INDSET}$

$$(x \vee y \vee z) \wedge (x \vee \bar{y} \vee y) \wedge (u \vee y \vee \bar{z}) \wedge (z \vee \bar{x} \vee u) \wedge (\bar{x} \vee \bar{y} \vee \bar{z})$$



$3\text{SAT} \leq_p \text{INDSET}$

$$(x \vee y \vee z) \wedge (x \vee \overline{y} \vee y) \wedge (u \vee y \vee \overline{z}) \wedge (z \vee \overline{x} \vee u) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$$

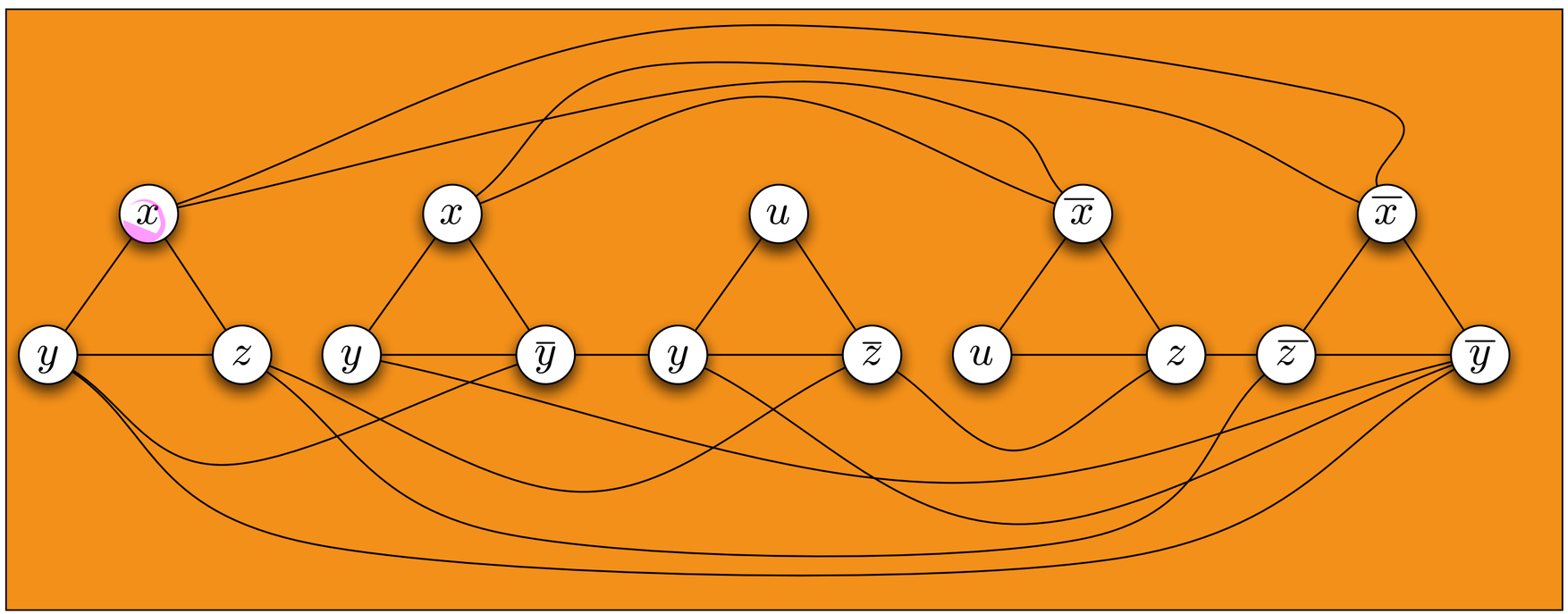


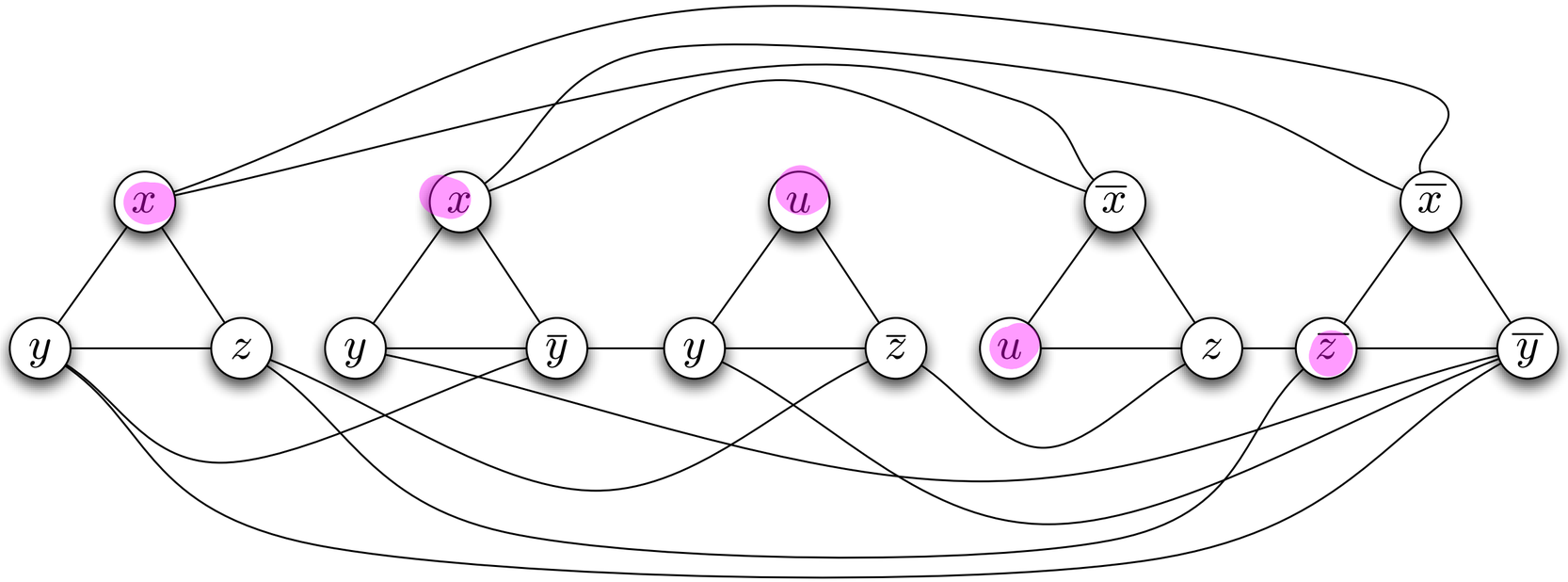
k clausy

$$(x \vee y \vee z) \wedge (x \vee \overline{y} \vee y) \wedge (u \vee y \vee \overline{z}) \wedge (z \vee \overline{x} \vee u) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$$

ϕ

(G, k)





$\exists$ an ind set if size k

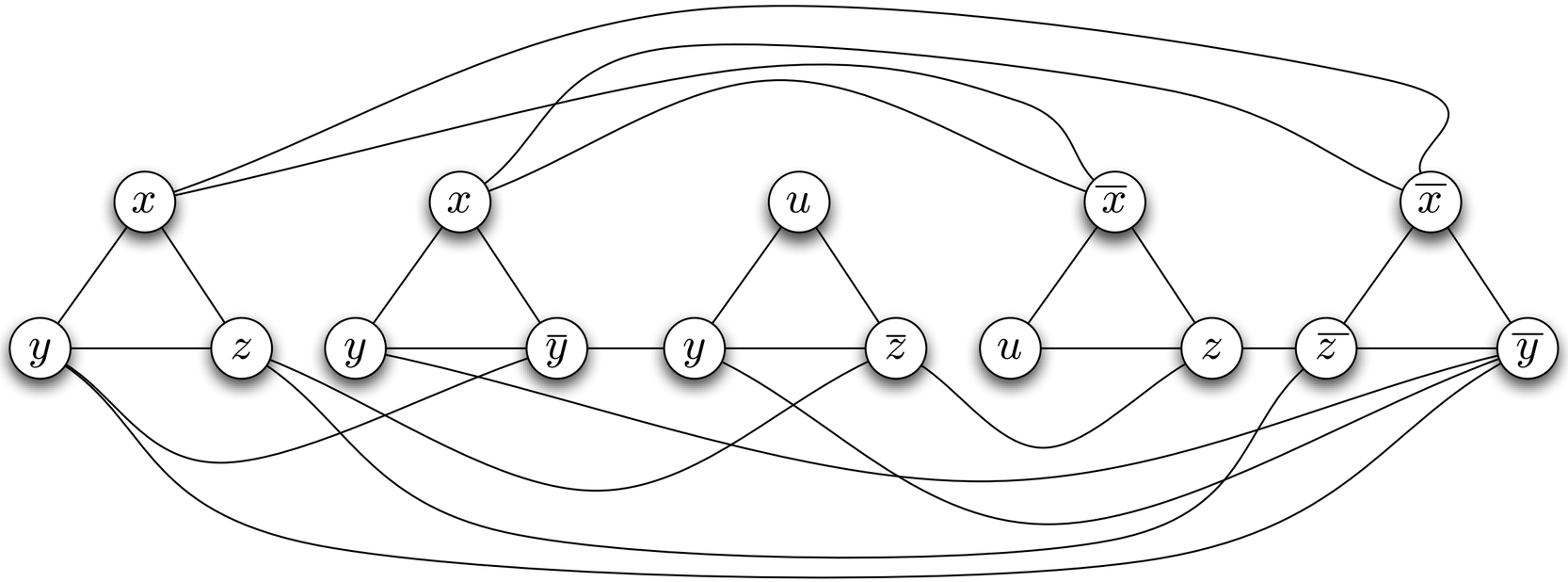
$\phi \in$ SAT $\Rightarrow$ $\exists$ a satisfying assignment A for ϕ,

$\Rightarrow$ each clause has a true variable (can be ≥ 1)

pick one true variable per clause.

Let $S = \{$ ↑ set $\}$. $|S| = k$. b/c there are k clauses.

AND if a variable is clause i is selected, then its negation is false, & not selected. 1 node per triangle



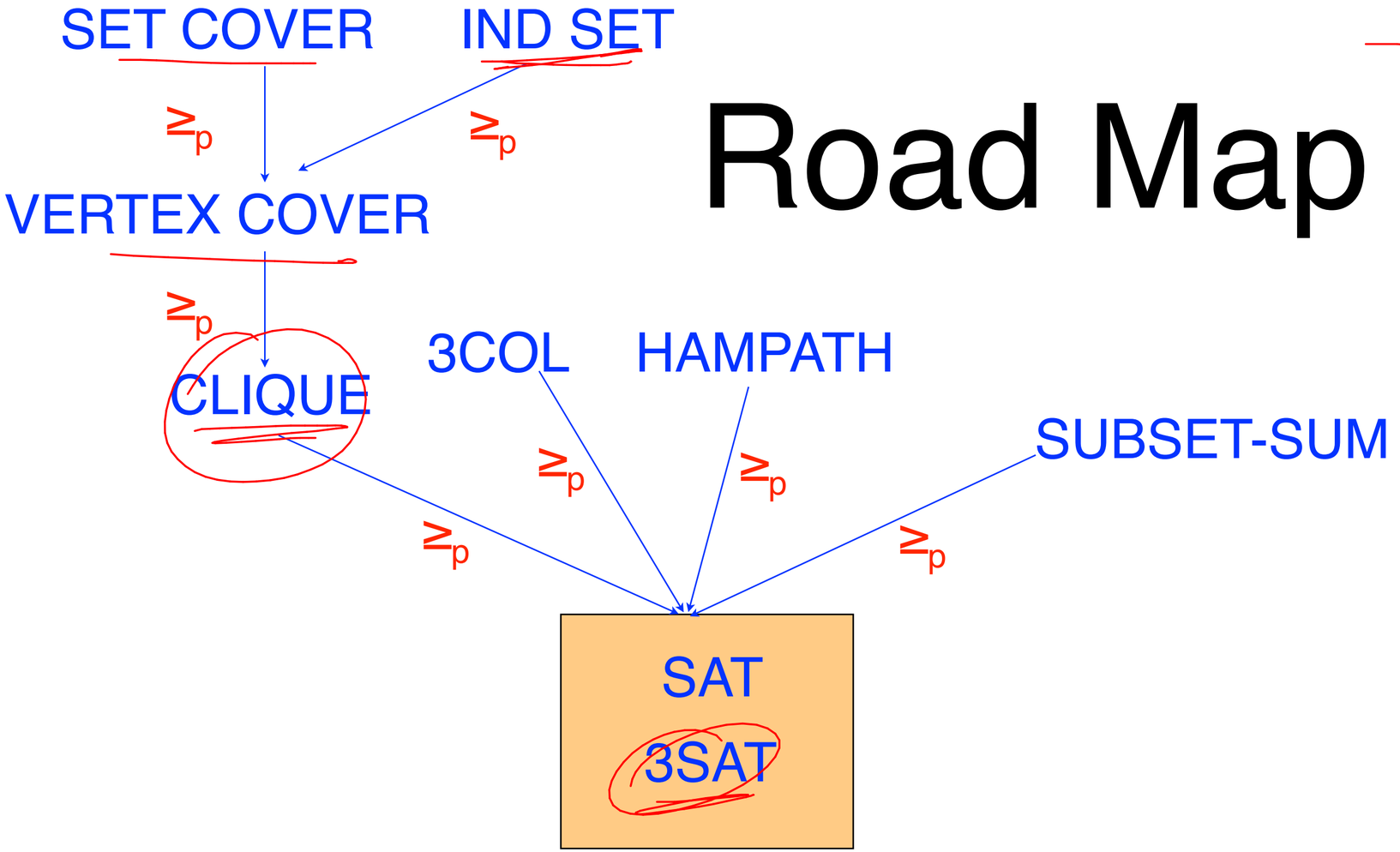
$(G, k) \in$ INDSET $\Longrightarrow$ By definition, only 1 node per triangle is selected. Make that variable true. Assign any unassigned (positive) variable to be T.

① The resulting A assigns every variable.

② The assignment A is consistent. If A(x) = true, then no vertex for $\overline{x}$ is in the INDset. $\Rightarrow$ $\overline{x}$ is assigned F.

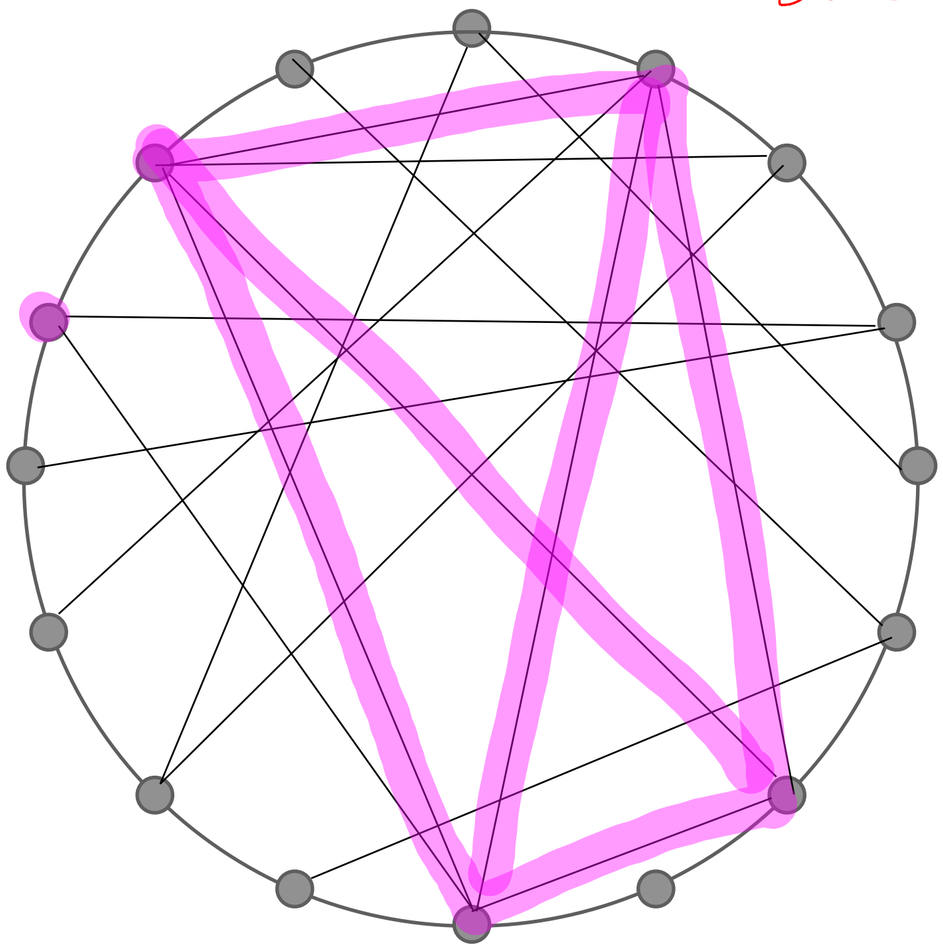
③ ϕ is satisfied by A. b/c each clause is SATISFIED.

Road Map


SET COVER
IND SET
≤p
≤p
VERTEX COVER
≤p
CLIQUE
3COL
HAMPATH
SUBSET-SUM
≤p
≤p
≤p
≤p
SAT
3SAT

clique

Social graph



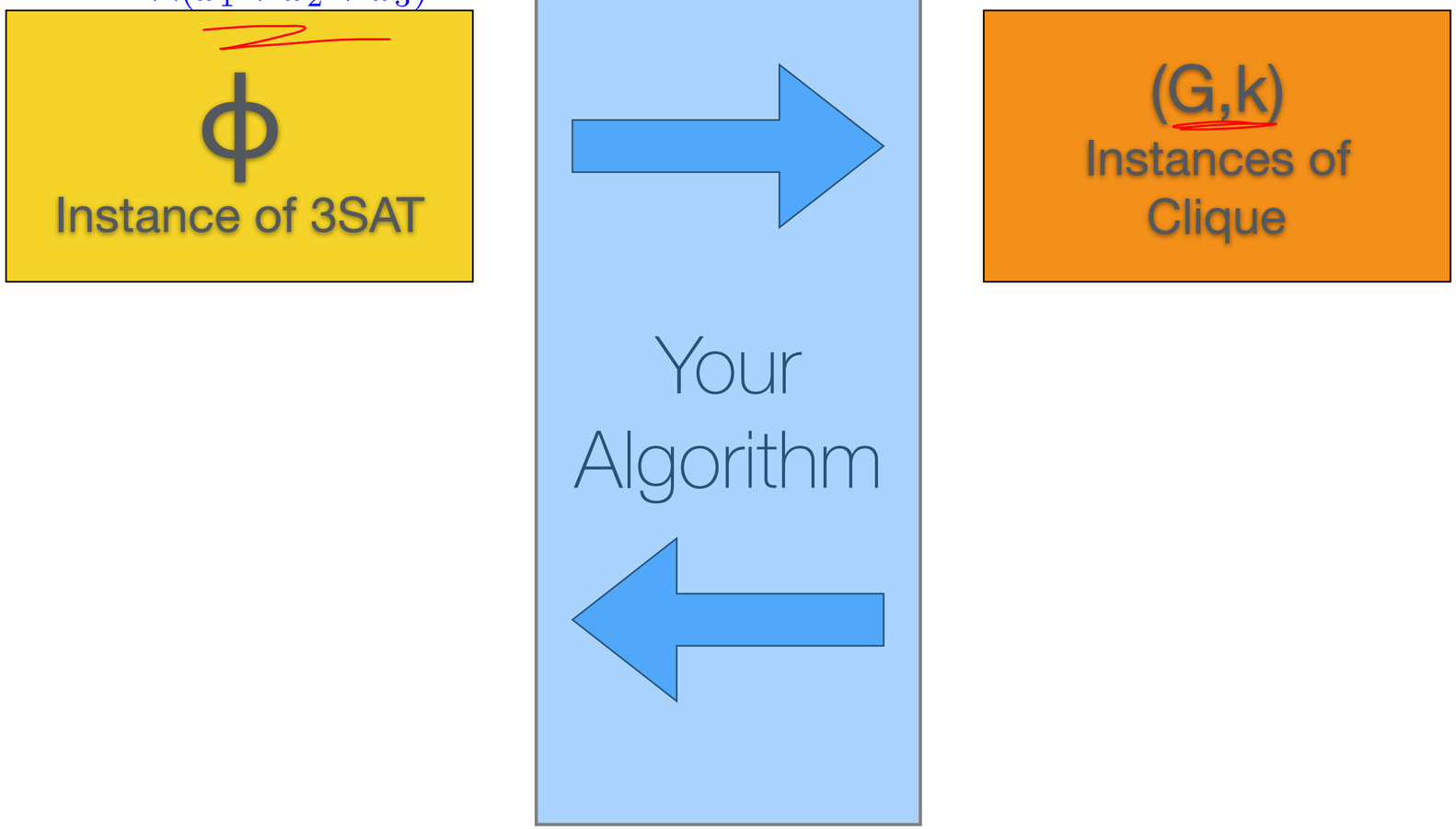
clique = { G, k , : determine if G has a clique of size k

→ a complete graph on k nodes
"all k vertices are connected to each other }

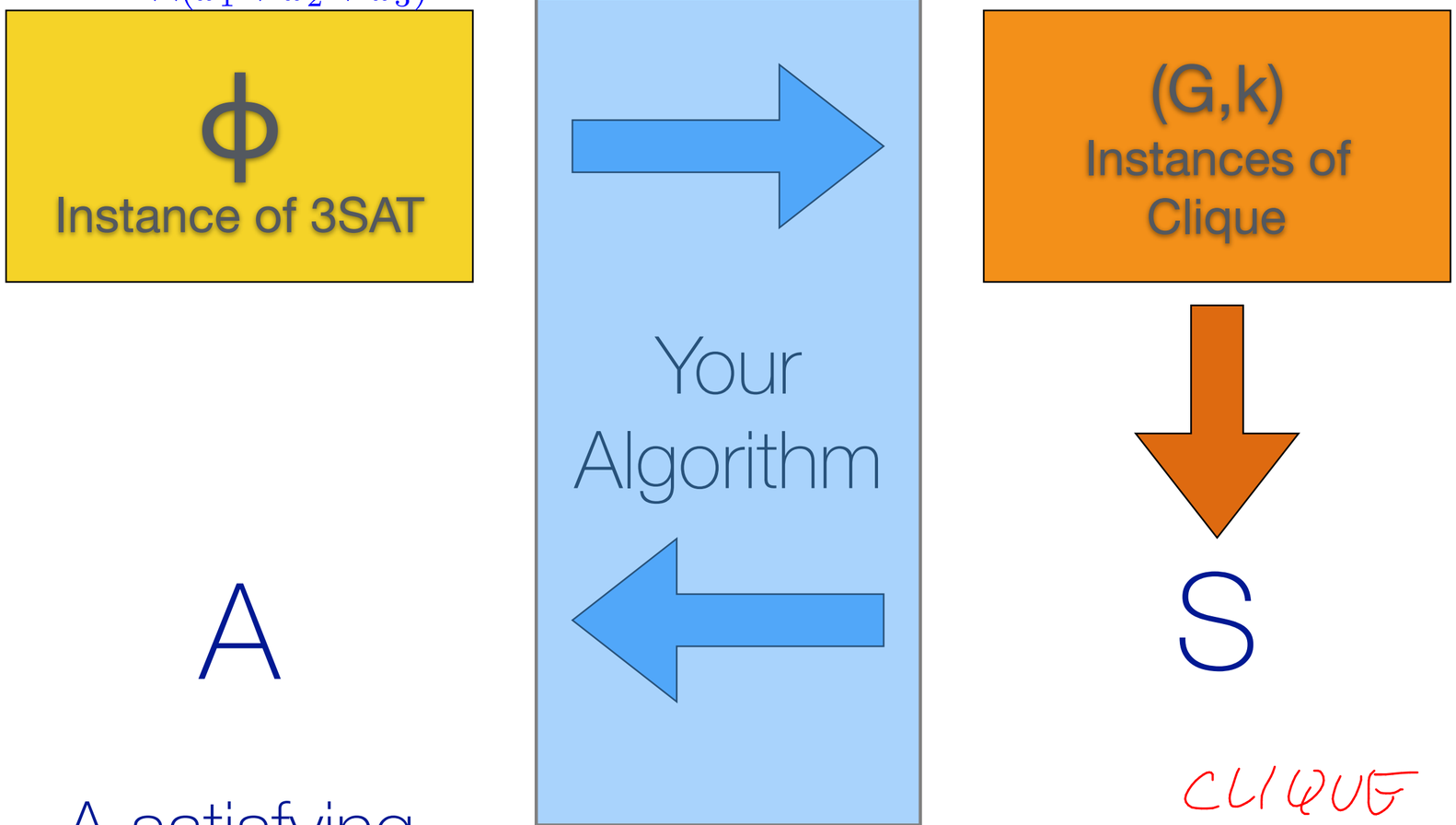
3SAT ≤ CLIQUE

"Clique is as hard as 3SAT upto polytime"

$$\phi = \begin{array}{l}(x_1 \vee x_2 \vee x_3) \\ \wedge(\overline{x_1} \vee \overline{x_2} \vee x_3) \\ \wedge(\overline{x_1} \vee x_2 \vee \overline{x_3})\end{array}$$



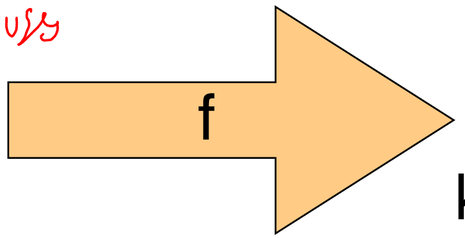
$$\phi = \begin{array}{l}(x_1 \vee x_2 \vee x_3) \\ \wedge(\overline{x_1} \vee \overline{x_2} \vee x_3) \\ \wedge(\overline{x_1} \vee x_2 \vee \overline{x_3})\end{array}$$



CLIQUE

formula

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$



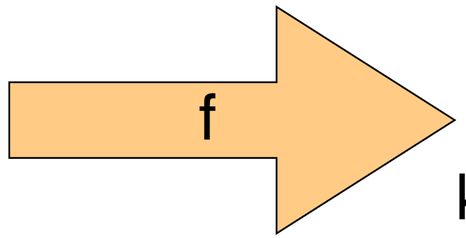
(Graph, k)

k = # clauses

CLIQUE

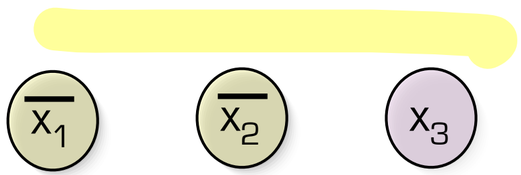
formula

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$



Graph, k

k = # clauses

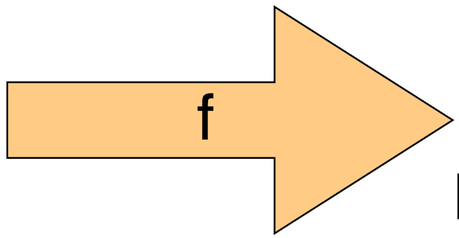


Create 3 nodes/clause

CLIQUE

formula

$$\phi = \begin{array}{l}(x_1 \vee x_2 \vee x_3) \\ \wedge(\overline{x_1} \vee \overline{x_2} \vee x_3) \\ \wedge(\overline{x_1} \vee x_2 \vee \overline{x_3})\end{array}$$

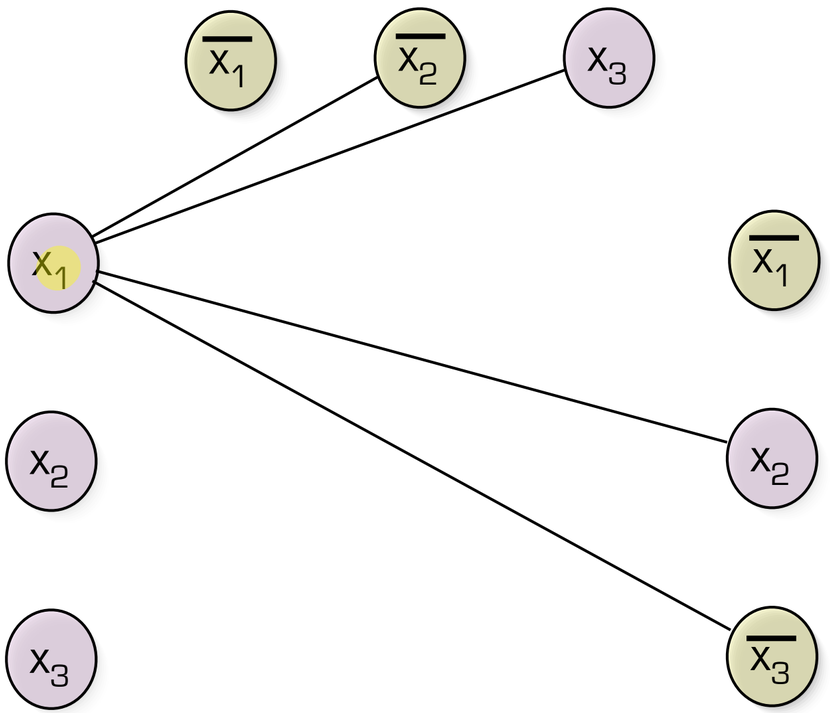


Graph, k

k = # clauses

Create 3 nodes/clause

Connect nodes to
"non-opposites"



CLIQUE

formula

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

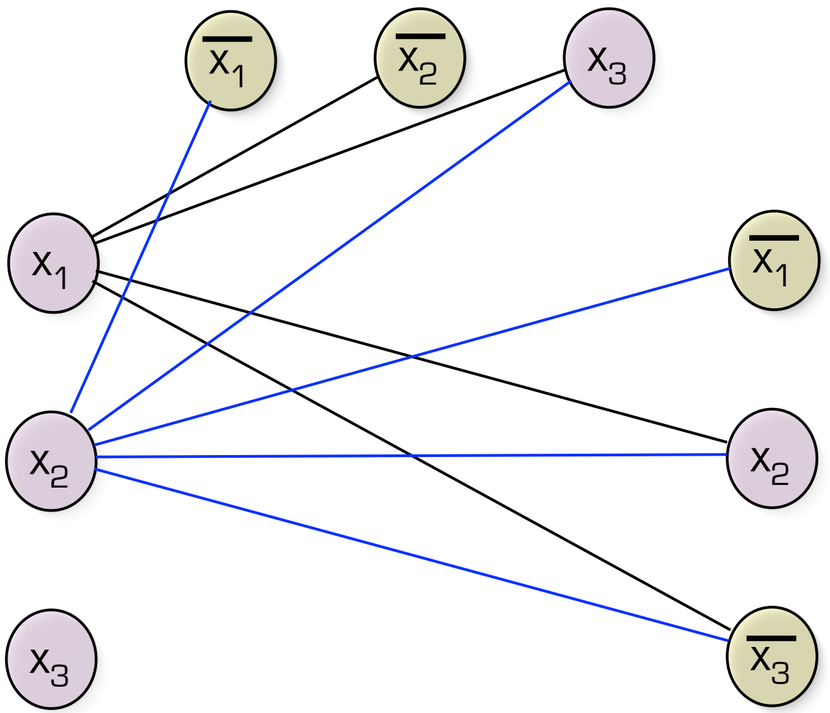
f

Graph, k

k = # clauses

Create 3 nodes/clause

Connect nodes to
"non-opposites"



CLIQUE

formula

$$\phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

f

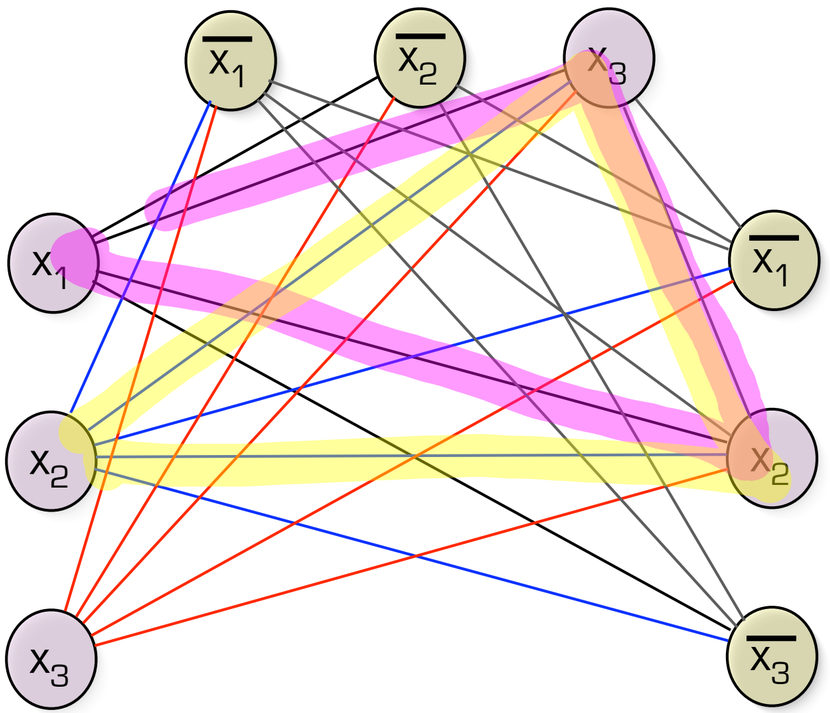
Graph, k

k = # clauses

order $3k \cdot V$ time

Create 3 nodes/clause

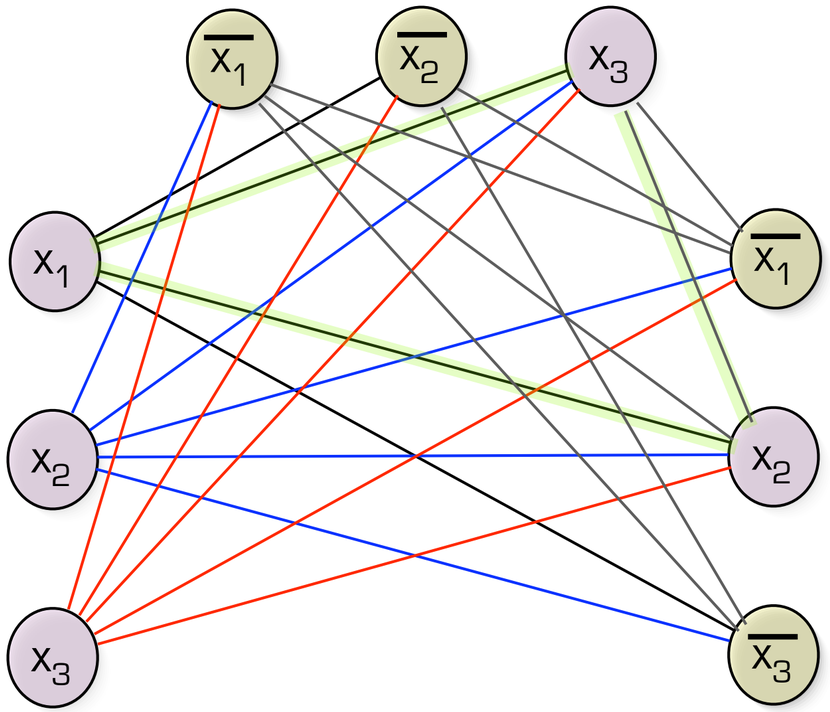
Connect nodes to "non-opposites"



if G has a k clique

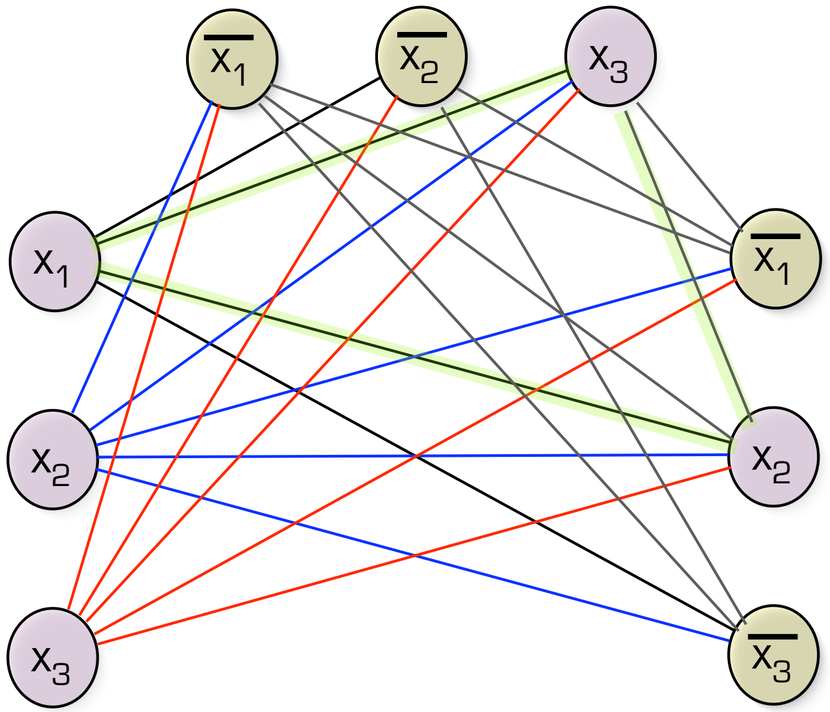
CLIQUE

$$\phi = \begin{array}{l} (x_1 \vee x_2 \vee x_3) \\ \wedge(\overline{x_1} \vee \overline{x_2} \vee x_3) \\ \wedge(\overline{x_1} \vee x_2 \vee \overline{x_3}) \end{array}$$



CLIQUE

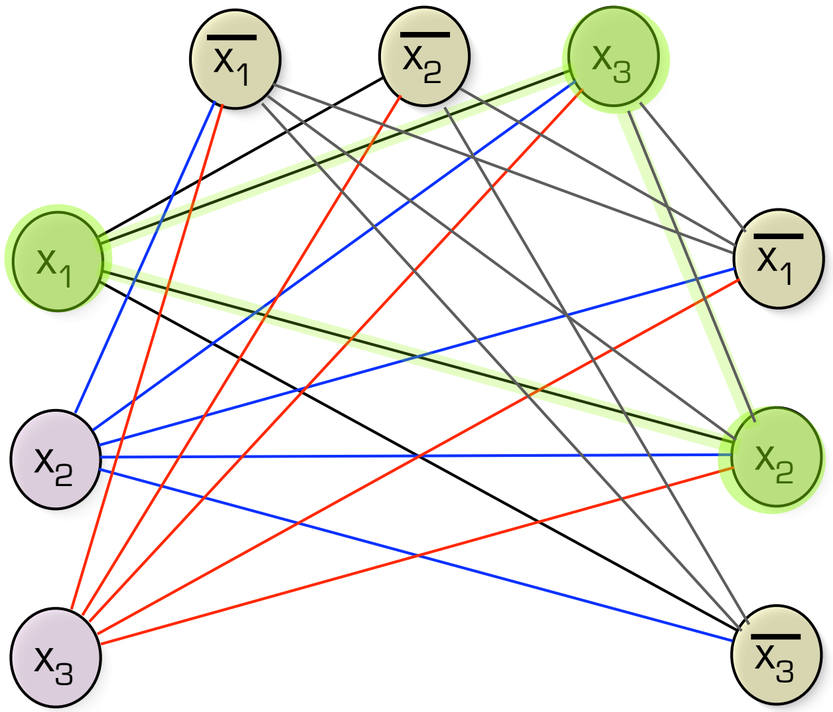
$$\phi = \begin{array}{l} (x_1 \vee x_2 \vee x_3) \\ \wedge(\overline{x_1} \vee \overline{x_2} \vee x_3) \\ \wedge(\overline{x_1} \vee x_2 \vee \overline{x_3}) \end{array}$$



Satisfying assignment = 1 var/clause

CLIQUE

$$\phi = \begin{array}{l} (x_1 \vee x_2 \vee x_3) \\ \wedge(\overline{x_1} \vee \overline{x_2} \vee x_3) \\ \wedge(\overline{x_1} \vee x_2 \vee \overline{x_3}) \end{array}$$

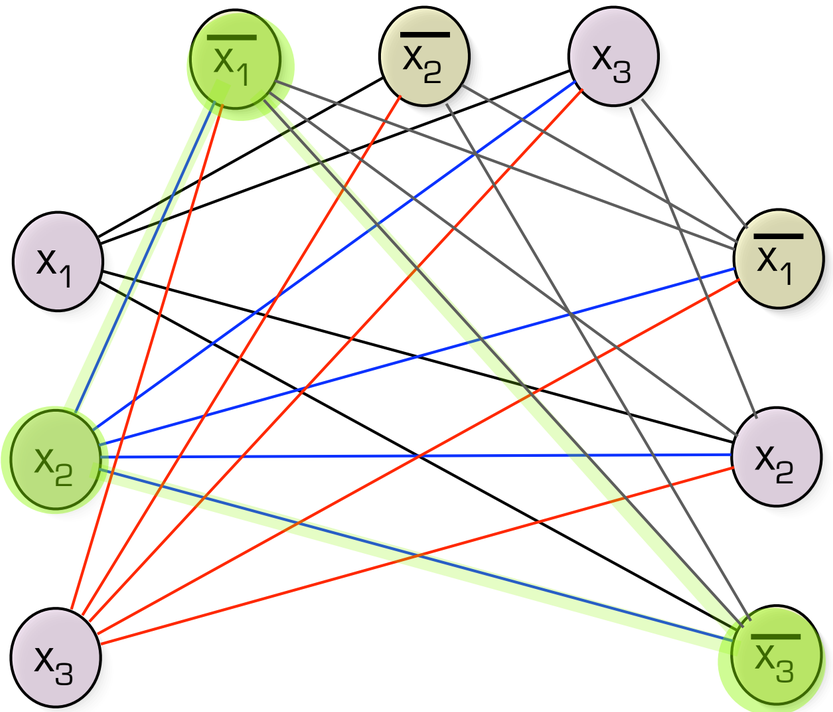


Satisfying assignment = 1 var/clause

k “non-opposite” connected nodes

CLIQUE

$$\phi = \begin{array}{l} (x_1 \vee x_2 \vee x_3) \\ \wedge(\overline{x_1} \vee \overline{x_2} \vee x_3) \\ \wedge(\overline{x_1} \vee x_2 \vee \overline{x_3}) \end{array}$$

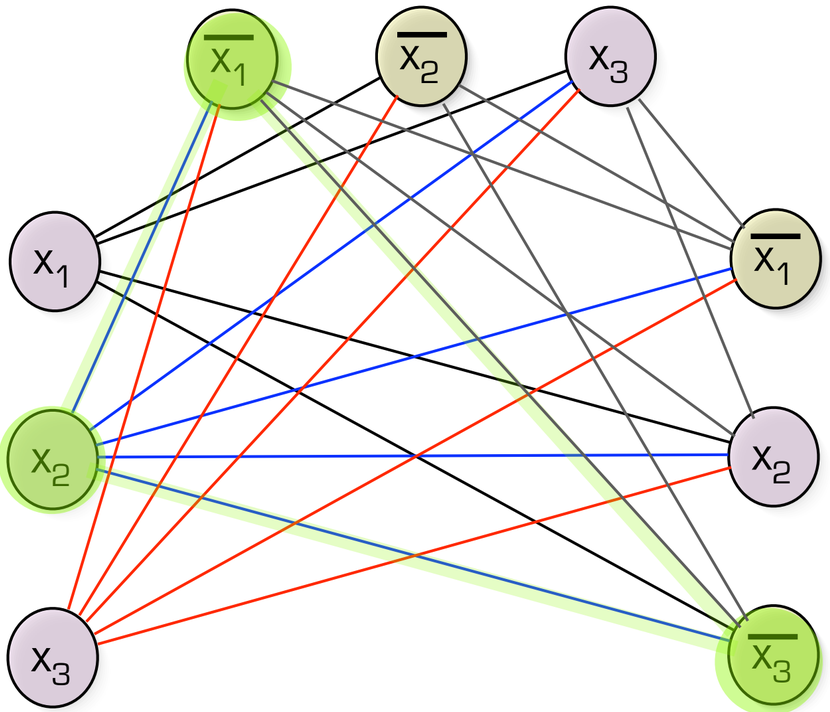


k-clique

1 node/clause is true

CLIQUE

$$\phi = \begin{array}{l}(x_1 \vee x_2 \vee x_3) \\ \wedge(\overline{x_1} \vee \overline{x_2} \vee x_3) \\ \wedge(\overline{x_1} \vee x_2 \vee \overline{x_3})\end{array}$$



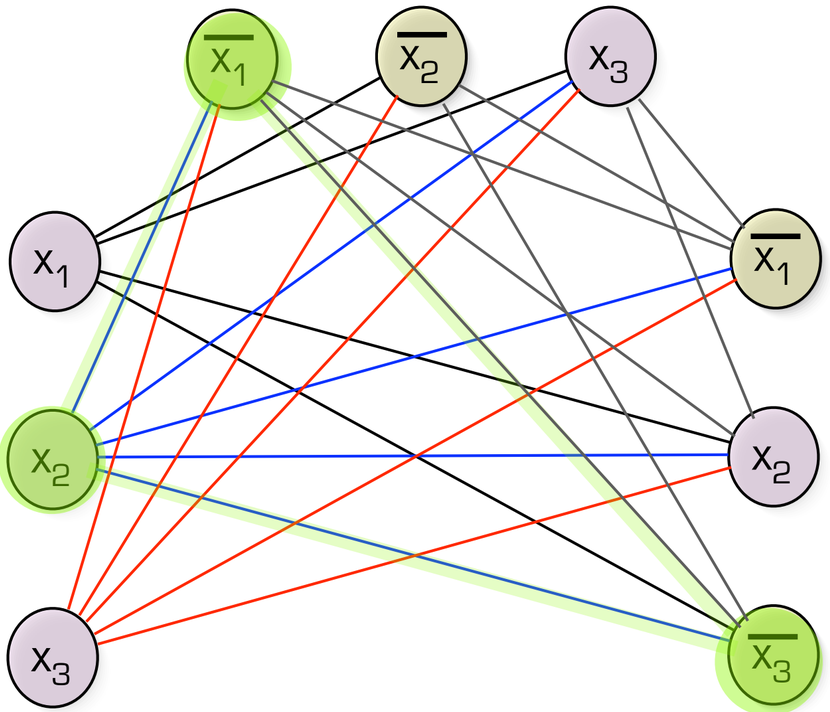
k-clique

1 node/clause is true

Satisfying assignment

CLIQUE

$$\phi = \begin{array}{l}(x_1 \vee x_2 \vee x_3) \\ \wedge(\overline{x_1} \vee \overline{x_2} \vee x_3) \\ \wedge(\overline{x_1} \vee x_2 \vee \overline{x_3})\end{array}$$



$$\phi \in SAT \quad \Leftrightarrow \quad f(\phi) \in CLIQUE$$

Theory of NP

Languages

$L = \{$ sets on instance $\}$

$x \in L$??

DEF OF NP

a language L belongs to the class NP iff
there exists a polynomial time algorithm A
and a constant c such that

$$L = \{x \in \{0,1\}^* \mid \exists y \in \{0,1\}^{|x|^c} \text{ s.t. } A(x,y) = 1\}$$

efficiently test whether a SOLUTION IS TRUE.

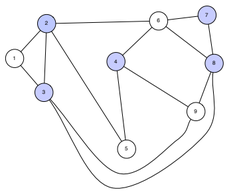
NP-Completeness

A language L is NP-Complete if

1. $L \in NP$
2. $\forall A \in NP, A \leq_p L$

"L is among the hardest NP problems"

WHY IS VC IN NP?



vertexcover(G,k)

COOK-LEVIN THEOREM

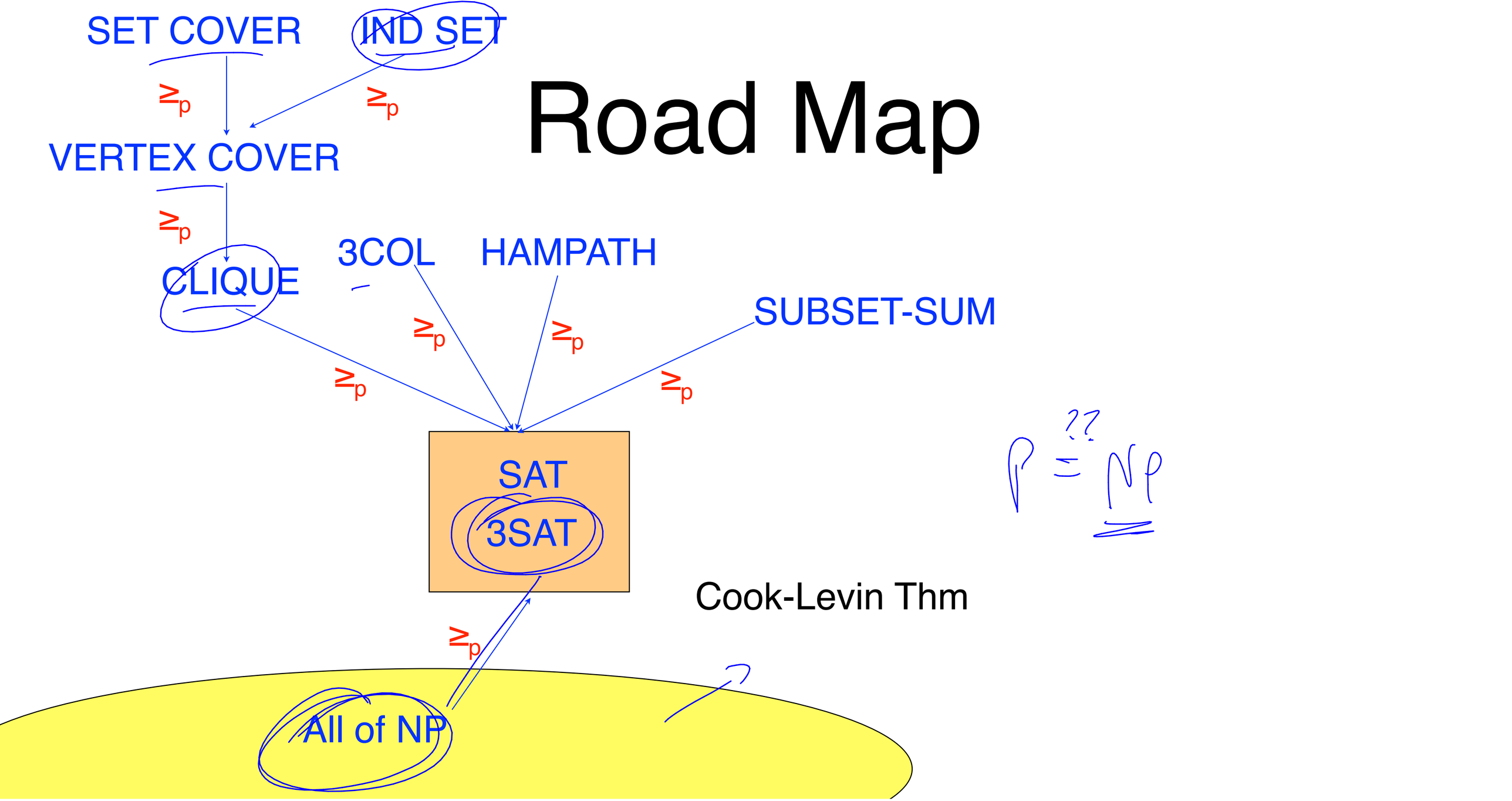
WHAT IS THE HARDEST PROBLEM IN NP?

Cook-Levin theorem

$$\forall L \in \mathrm{NP}$$

$$L \leq_f \mathbf{3SAT}$$

Road Map


SET COVER
IND SET
≤p
≤p
VERTEX COVER
≤p
CLIQUE
3COL
HAMPATH
SUBSET-SUM
≤p
≤p
≤p
≤p
SAT
3SAT
≤p
All of NP
Cook-Levin Thm
P = ?? NP