Computational Tractability

Plan: Explore the space of computationally hard problems to arrive at a mathematical characterization of a large class of them.

Technique: Compare relative difficulty of different problems.

Loose definition: If problem X is at least as hard as problem Y, it means that if we could solve X, we could also solve Y.

Max flow is as hard as feasible circulation

→ harder

feasible circulation ⇒ max flow

If I can solve max flow then I can also solved feasible circulation
Hence max flow is as hard as feasible circulation

Formal definition:

poly time reducible

$Y \leq_p X$ (Y is polynomial time reducible to X)

if Y can be solved using a polynomial number of standard computational steps plus a polynomial number of calls to a blackbox that solves X.

eg: addition, assignment, etc

non standard computation
- sorting
sorting depends upon size of array
hence no. of cycles change

requires fix number of cpu cycles

Suppose $Y \leq_p X$, if X can be solved in polynomial time, then Y can be solved in polynomial time.

Suppose $Y \leq_p X$, if Y cannot be solved in polynomial time, then X cannot be solved in polynomial time

Independent Set

Def. In a graph $G=(V,E)$, we say that a set of nodes $S \subseteq V$ is "independent" if no two nodes in S are joined by an edge.

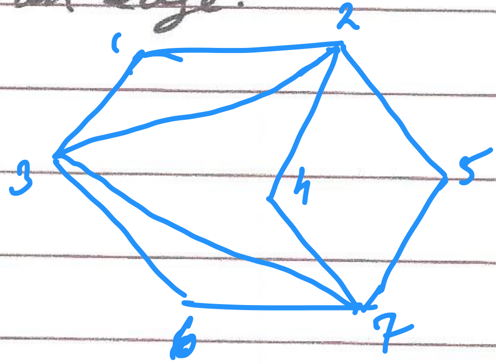
Independent set

$\{1,4,5,6\}$

$\{1\}$

$\{1,4\}$

$\vdots$



Independent set problem

- Find the largest independent set in graph G.

(opt version)

- Given a graph G, and a number k does G contain an independent set of size at least k?

(decision version) → Requires Yes or No answer

Opt prob is atleast as hard as decision prob

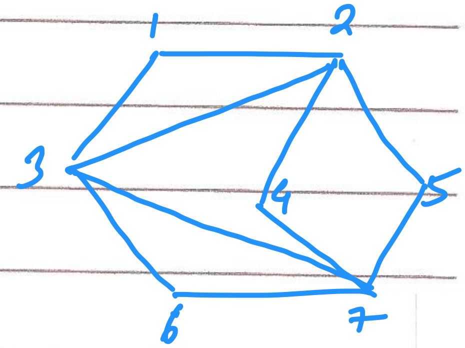
Vertex Cover

Def. Given a graph $G=(V,E)$, we say that a set of nodes $S \subseteq V$ is a vertex cover if every edge in E has at least one end in S.

vertex cover

$\{2,4,7\}$

$\{1,2,3,4,5,6,7\}$



Vertex Cover problem

- Find the smallest vertex cover set in G.

(opt version)

- Given graph G and number k, does G contain a vertex cover set of size atmost $\underline{k}$?

(decision version)

FACT: let $G=(V,E)$ be a graph, then S is an independent set if and only if its complement $V-S$ is a vertex cover set.

Proof: A) First suppose that S is an independent set

Case 1 : U is in S and V is not
V-S will have V and not U

Case 2 : V is in S and U is not
V-S will have U and not V

Case 3 : Neither V nor U in S
V-S will have both V and U

B) Suppose that V-S is a vertex cover set then S is independent set

Claim: Indep. set $\leq_p$ vertex cover

Proof: If we have a blackbox to solve vertex cover, we can decide if G has an independent set of size at least $\underline{k}$, by asking the blackbox if G has a vertex cover of size atmost $n-k$

1 vertex cover is atleast as hard as indp set problem

Claim: Vertex Cover $\leq_p$ Indep set

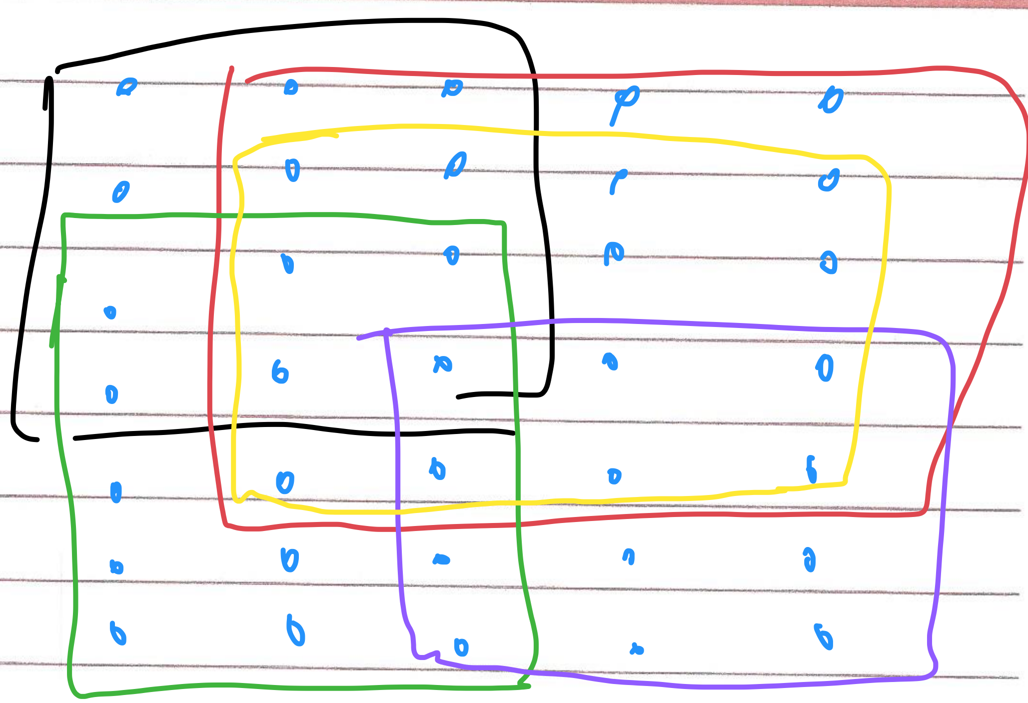
Proof: If we have a blackbox to solve independent set, we can decide if G has a vertex cover set of size at most $\underline{k}$, by asking the blackbox G has an independant of size atleast $n-k$

2 Indp set is atleast as hard as vertex cover

Hence it means they are within the same complexity class from 1 and 2

Set Cover Problem

Given a set U of $\underline{n}$ elements, a collection $S_1, S_2, \ldots S_m$ of subsets of U, and a number $\underline{k}$, does there exist a collection of at most $\underline{k}$ of these sets whose union is equal to all of U.



set cover problem

vertex cover problem

Claim: Vertex Cover $\leq_p$ Set Cover

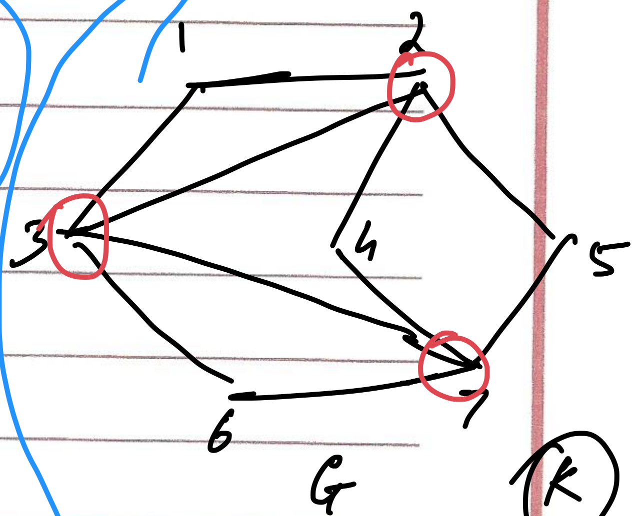
$S_1 = \{(1,2)\ (1,3)\}$

$S_2 = \{(1,2)\ (2,4)\ (2,5)\ (2,3)\}$

S_3

$\vdots$

S_7



k

k

Need to show that G has a vertex cover of size k, iff the corresponding set cover instance has $\underline{k}$ sets whose union ~~is~~ contains ~~equal to~~ all edges in G.

Proof:

A) If I have a vertex cover set of size $\leq k$ in G, I can find a collection of $\leq k$ sets whose union contains all edges in G.

eg: Given $\{2,3,7\}$ as vertex cover set

$(S_2 \cup S_3 \cup S_7)$ will cover all edges

set cover solution

B) If I have k sets whose union contains all edges in G, I can find a vertex cover set of size k in G.

Reduction Using Gadgets

Not graph problem to graph problem

Here we use gadgets

- Given $\underline{n}$ Boolean variables $x_1 \ldots x_n$, a <u>clause</u> is a disjunction of terms $t_1 \vee t_2 \vee \ldots t_l$ where $t_i \in \{x_1 \ldots x_n, \bar{x}_1 \ldots \bar{x}_n\}$

- A <u>truth assignment</u> for X is an assignment of values $\underline{0}$ or $\underline{1}$ to each x_i.

- An assignment <u>satisfies</u> a clause C if it causes $\underline{C}$ to evaluate to 1.

- An assignment satisfies a collection of clauses if

$$C_1 \wedge C_2 \wedge \ldots \quad C_k$$

~~at~~ evaluates to 1.

example :

$$(x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (x_2 \vee \bar{x}_3)$$

✗ $x_1 = 1$ $x_2 = 1$ $x_3 = 1$

if this a satisfied consignment?

No

✓ if $x_1 = 0$ $x_2 = 0$ $x_3 = 0$ (satisfying consignment)

✓ $x_1 = 1$ $x_2 = 0$ $x_3 = 0$ (satisfying consignment)

Problem Statement: Given a set of clauses $C_1 \ldots C_k$ over a set of variables $X = \{x_1 \ldots x_n\}$ does there exist a satisfying truth assignment?

satisfiability problem (SAT)

Problem statement: Given a set of clauses $C_1 \ldots C_k$ each of length $\underline{3}$ over a set of variables $X = \{x_1 \ldots x_n\}$ does there exist a satisfying truth assignment?

3 SAT problem

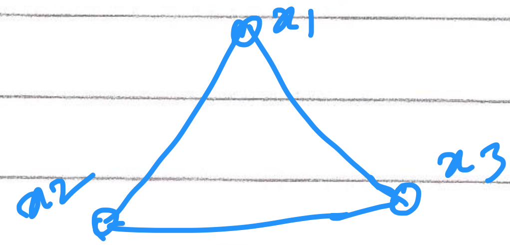
non graph problem

graph problem

Claim: 3SAT $\leq_p$ Independent Set

Plan: Given an instance of 3SAT with $\underline{k}$ clauses, build a graph G that has an indep. set of size $\underline{k}$ iff the 3SAT instance is satisfiable.

$(x_1 \vee x_2 \vee x_3)$

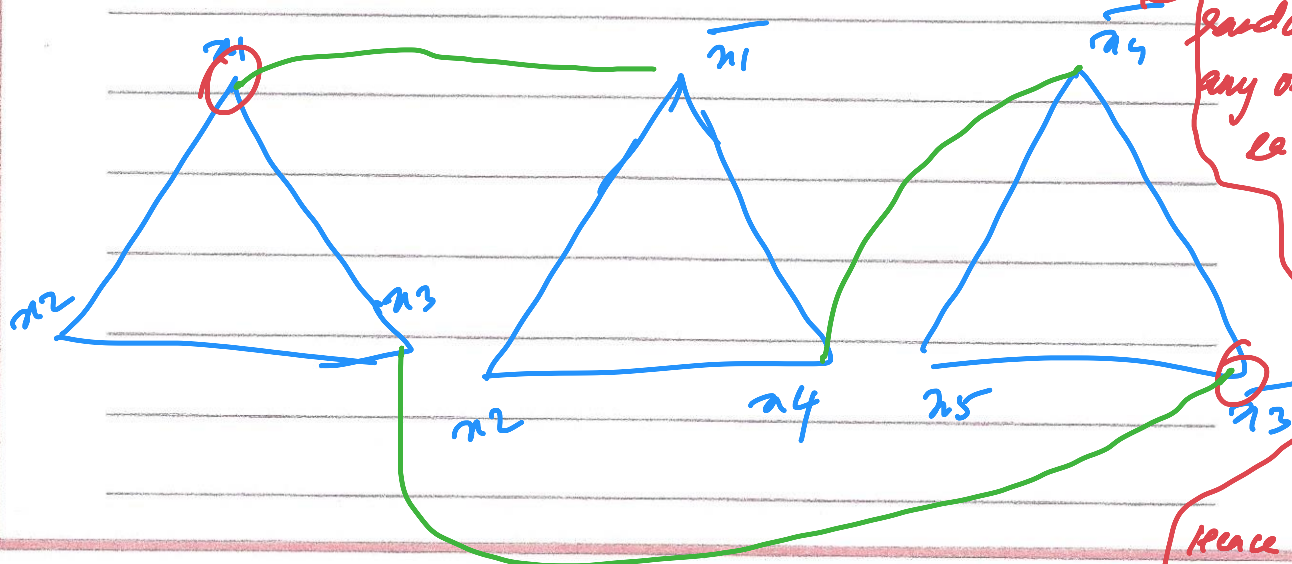


example

$C_1 = (x_1 \vee x_2 \vee x_3)$

$C_2 = (\bar{x}_1 \vee x_2 \vee x_4)$

$C_3 = (\bar{x}_4 \vee x_5 \vee \bar{x}_3)$



only one node from each graph can be independent set

So we can randomly choose any one from each graph

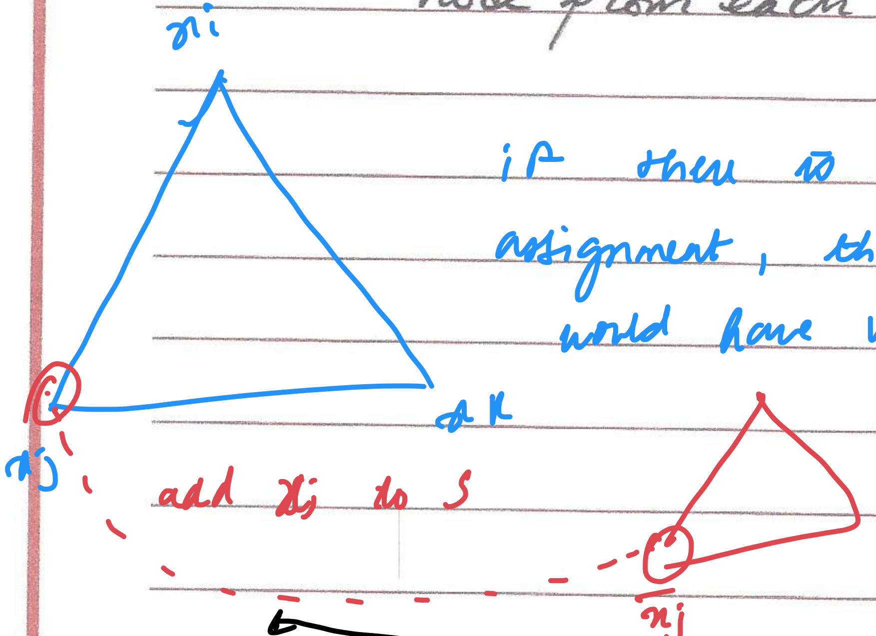
prob IF we choose x_1 and $\bar{x}_1$ no way $x_1 \wedge \bar{x}_1$

Hence connect them so only one of them can be chosed

Claim: The 3-SAT instance is satisfiable iff the graph G has an independent set of size $\underline{k}$.

Proof: A) If the 3-SAT instance is satisfiable, then there is at least one node label per triangle that evaluates to 1.

Let S be a set containing one such node from each triangle



if there is a satisfying truth assignment, then atleast one of x_i, x_j, x_k would have value of 1

only problem could be if we add x_j and $\bar{x}_j$ to set S

but this is never possible as x_j and $\bar{x}_j$ both can never have value of 1

Hence this is not a problem

one node from every gadget in S

B) Suppose G has an independent set S of size at least k.

if x_i appears as a label in S then **set $x_i = 1$**

if $\bar{x}_i$ appears as a label in S Then **set $x_i = 0$**

if neither x_i nor $\bar{x}_i$ appear as a label in S, then **it does not matter**

Efficient Certification

To show efficient certification:

1. Polynomial length certificate

2. Polynomial time certifier

Efficient certification

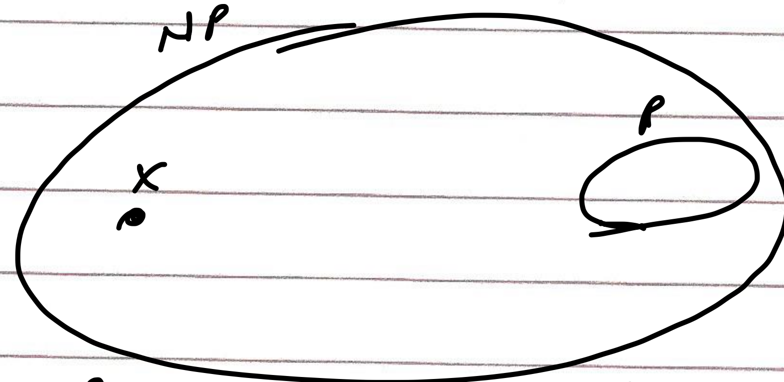
3-SAT Certificate t is an assignment of truth values to variables (x_i)

Certifier: evaluate the clauses. If all of them evaluate to 1 then it answers yes.

Indep set Certificate t is a set of nodes of size at least k in G.

Certifier: check each edge to make sure no edges have both ends in the set ✓
check size of the set $\geq k$ ✓
no repeating nodes ✓

class NP is the set of all problems for which there exists an efficient certifier



Can P = NP? we dont know!

if x is in NP and for all $Y \in NP$

$Y \le_p X$, then x must be hardest problem in NP

3-SAT has been proven to be such a problem

hardest problem

Such a problem is called NP-complete

hardest NP problem

Transitivity

If $Z \le_p Y$ and $Y \le_p X$

then $\boxed{Z \le_p X}$

It has been proved 3 SAT is NP complete

3 SAT $\le_p$ indp set $\le_p$ vertex cover $\le_p$ set cover

$\therefore$ indp set is as hard as 3 SAT

$\therefore$ independent is also NP complete

$\therefore$ vertex cover is also NP complete

$\therefore$ set cover is also NP complete

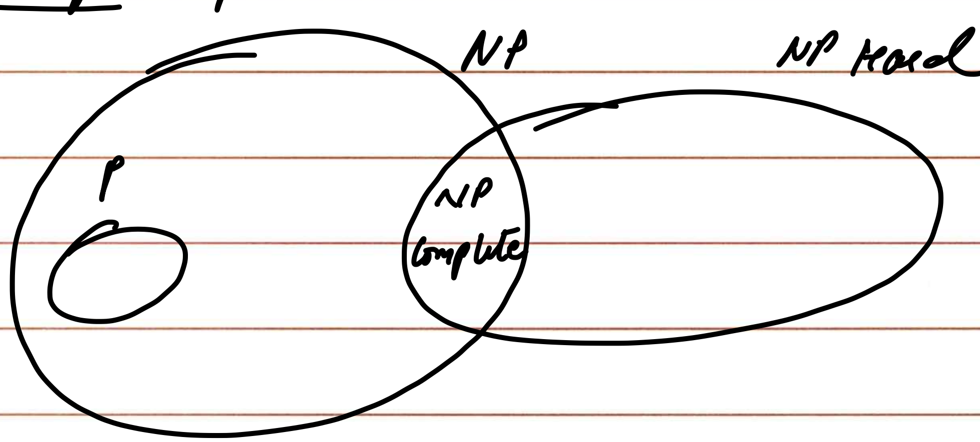
How to show problem is NP complete?

Basic strategy to prove a problem x is NP complete

1. We need to prove x is in class NP
2. Choose a problem Y that is known to be NP complete
3. Prove that $Y \le_p X$

This proves X is atleast as hard as Y
and Y is NP complete
∴ x is NP complete

NP Hard problems are atleast as hard as **NP complete** problems



If from above steps

if we only do step 3

i.e $Y \leq_p X$

we prove X is NP-hard

only if X is NP

and NP hard

then X is NP complete

For NP problems, we try to find approximate solutions.

Discussion 10

1. Given the SAT problem from lecture for a Boolean expression in Conjunctive Normal Form with any number of clauses and any number of literals in each clause. For example,

$$(X_1 \vee \neg X_3) \wedge (X_1 \vee \neg X_2 \vee X_4 \vee X_5) \wedge \ldots$$

Prove that SAT is polynomial time reducible to the 3-SAT problem (in which each clause contains at most 3 literals.)

2. The *Set Packing* problem is as follows. We are given *m* sets $S_1, S_2, \ldots S_m$ and an integer *k*. Our goal is to select *k* of the *m* sets such that no selected pair have any elements in common. Prove that this problem is **NP**-complete.

3. The *Steiner Tree* problem is as follows. Given an undirected graph $G=(V,E)$ with nonnegative edge costs and whose vertices are partitioned into two sets, *R* and *S*, find a tree $T \subseteq G$ such that for every *v* in *R*, *v* is in *T* with total cost at most *C*. That is, the tree that contains every vertex in *R* (and possibly some in *S*) with a total edge cost of at most *C*.
Prove that this problem is **NP**-complete.

① SAT | 3 SAT

(x_1) | $(x_1 \vee x_1 \vee x_1)$

$(x_1 \vee x_2)$ | $(x_1 \vee x_2 \vee x_2)$

$(x_1 \vee x_2 \vee x_3)$ | $(x_1 \vee x_2 \vee x_3)$

$(x_1 \vee x_2 \vee x_3 \vee x_4)$ | $(x_1 \vee x_2 \vee s_1) \wedge (\bar{s}_1 \vee x_3 \vee x_4)$

suppose $x_2 = 1$ and $s_1 = 0$ $\therefore \left((x_1 \vee x_2 \vee s_1) \to T, \ (\bar{s}_1 \vee x_3 \vee x_4) \to T \right)$

$\therefore$ If we have value of any x_j we can set values for all x

$(x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5)$

$(x_1 \vee x_2 \vee s_1)(\bar{s}_1 \vee x_3 \vee s_2)(\bar{s}_2 \vee x_4 \vee x_5)$

if you give me satisfying truth assign on right it will give you satisfying truth assign on left

②

step1: Show the prob is in NP

polynomial length certificate: A set of K sets that have no elements in common

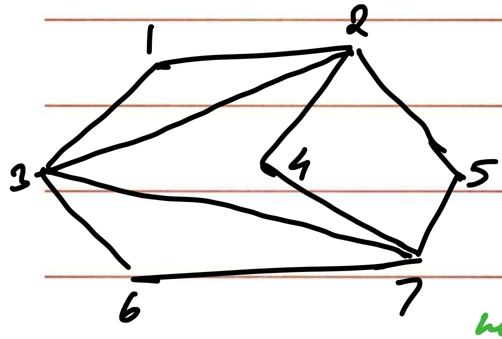
polynomial time certificate: check that sets have no element in common

step2: Choose a problem in NP complete that will hopefully lead to an easy reduction

Independent set

step3: Show that indp set $\leq_p$ set packing

start with an instance of indp set prob



G, K

while reduction we are only reducing in 1 direction

set packing — indp set

$S_1 = \{(1,2)\ (1,3)\}$

$S_2 = \{(1,2)\ (2,3)\ (2,4)\ (2,5)\}$

$\vdots$

S_7

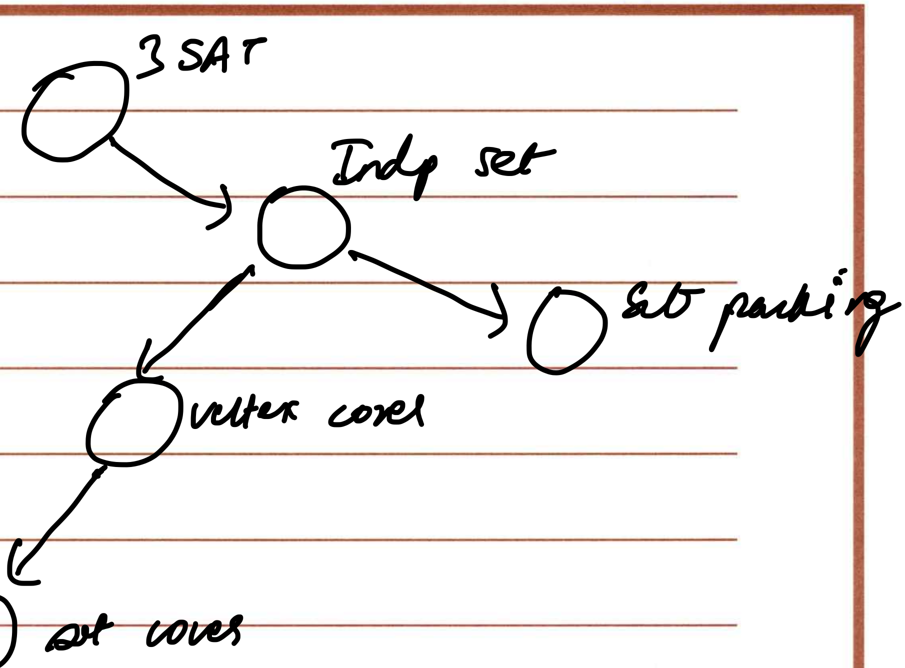
K will be same that goes in black box

If you give one indp set of size K, I can create K sets with no element in common

Given K sets with no elem in common, I can find indp set of size K

NOTE

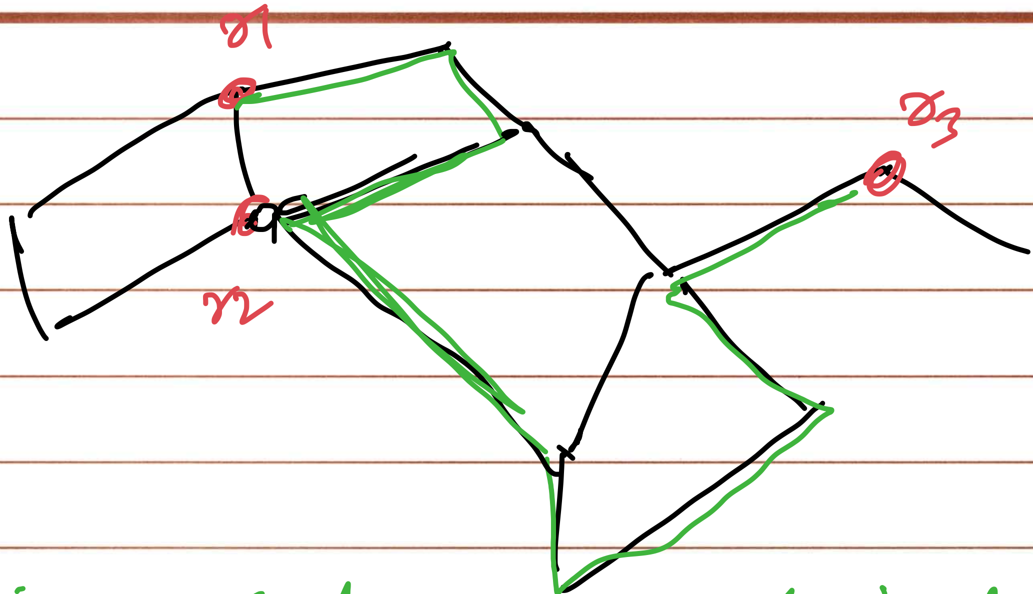
It has been shown that all NP complete problems can be converted to 3SAT



How to convert set packing → indp set

set packing – 3SAT — indp set

③



finding MST for whole graph is P

finding MST for a set of nodes is NP

Step 1 : Steiner Tree is NP

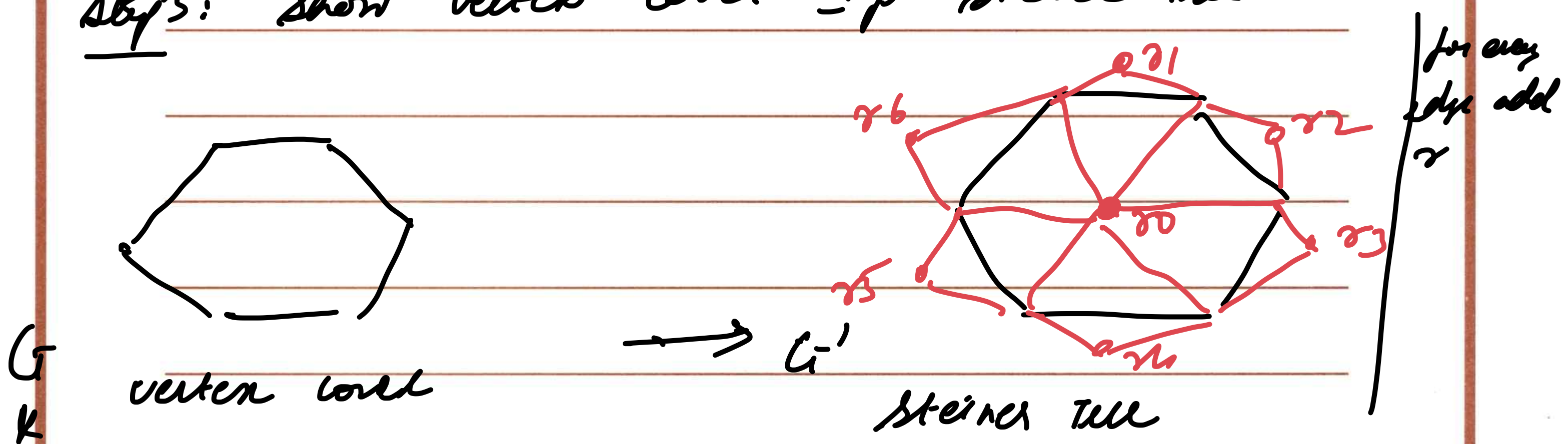
Certificate : Steiner Tree of cost $\leq C$

Certifier : - Check that T is a Tree
- check that tree covers all nodes in K
- Check that tree has cost $\leq C$

Step2: choose a problem in NP complete :

vertex Cover

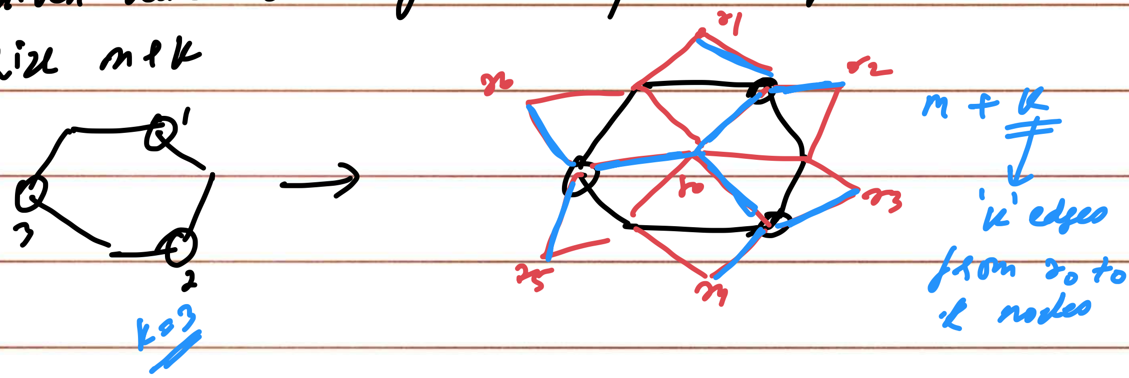
Step3: show vertex cover $\le_p$ steiner Tree



Is there a steiner tree of cost $\le$ M+K in G'?

PROOF

(A) Given vertex cover of size k, I can find steiner tree of size m+k



4) 20 pts

Suppose we have a variation on the 3-SAT problem called Min-3-SAT, where the literals are never negated. Of course, in this case it is possible to satisfy all clauses by simply setting all literals to true. But, we are additionally given a number k, and are asked to determine whether we can satisfy all clauses while setting at most k literals to be true. Prove that Min-3-SAT is NP-Complete.

(B) If you give me steiner tree of cost m+k, I can find vertex cover of size k

Just find nodes that have direct connection from s_0.

put all those nodes in vertex cover set of G

Step 1: Prove that Min-3SAT is in NP

Certificate: truth assignment with at most k literals set to true

Certifier: Evaluate each clause and if they all evaluate to 1, and at most k literals are set to true then answer yes.

Step 2: Choose a problem that we know is NP-Complete

Step 3: Prove that Vertex Cover $\leq_p$ Min-3SAT

Plan: Given an instance of vertex cover, we will create a set of clauses where there will be a satisfying truth assignment with at most $\underline{k}$ literals set to true iff there is a vertex cover of size ~~k~~ at most $\underline{k}$ in G.

Proof:

A) If I have a vertex cover of size at most $\underline{k}$ in G, I can create a satisfying truth assignment with at most $\underline{k}$ literals set to true, by

B) If I have a satisfying truth assignment with at most $\underline{k}$ literals set to true, I can create a vertex cover set by