Computational Tractability

Prob's for which we have pol. time sol's

unsolvable/undecidable

P

Ex.: Halting Prob.

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.

Formal definition:

$Y \leq_p X$ (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.

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 pol time.

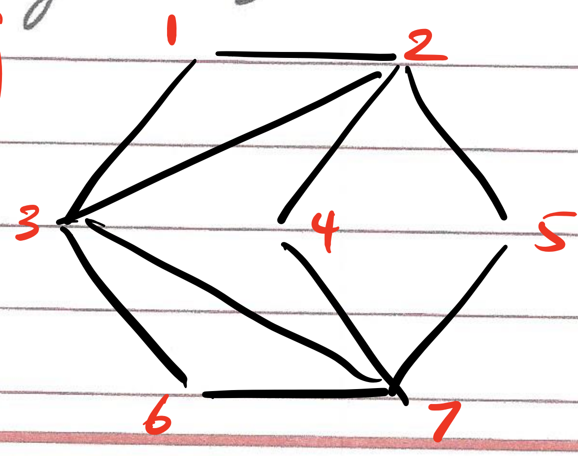
Independent Set

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.

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

$\{3,4,5\}$

$\{1\}$



Independent set problem

- Find the largest independent set in graph G.

  (opt. version)

- Given a graph G, and a no. k does G contain an indep set of size at least k?

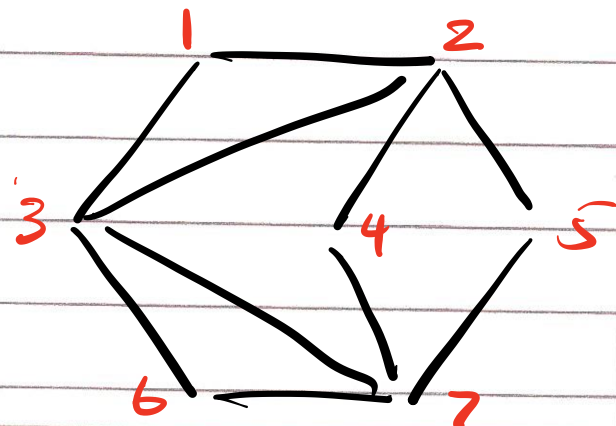
  (decision version)

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.

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

$\{2,3,7\}$



Vertex Cover problem

- Find the smallest vertex cover set in G.

(opt. version)

- Given a graph G and a no. $\underline{k}$ does G contain a vertex cover set of size at most $\underline{k}$?

(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

1 - U is in S, and V is not
$\Rightarrow$ $V-S$ $\rightarrow$ will have V and not U

2 - V is in S, and U is not
$\Rightarrow$ $V-S$ $\rightarrow$ will have U and not V

3 - Neither V nor U is in S
$\Rightarrow$ $V-S$ will have both V & U

B) Suppose $V-S$ is a vertex cover...

Claim: Indep. set $\leq_p$ vertex Cover

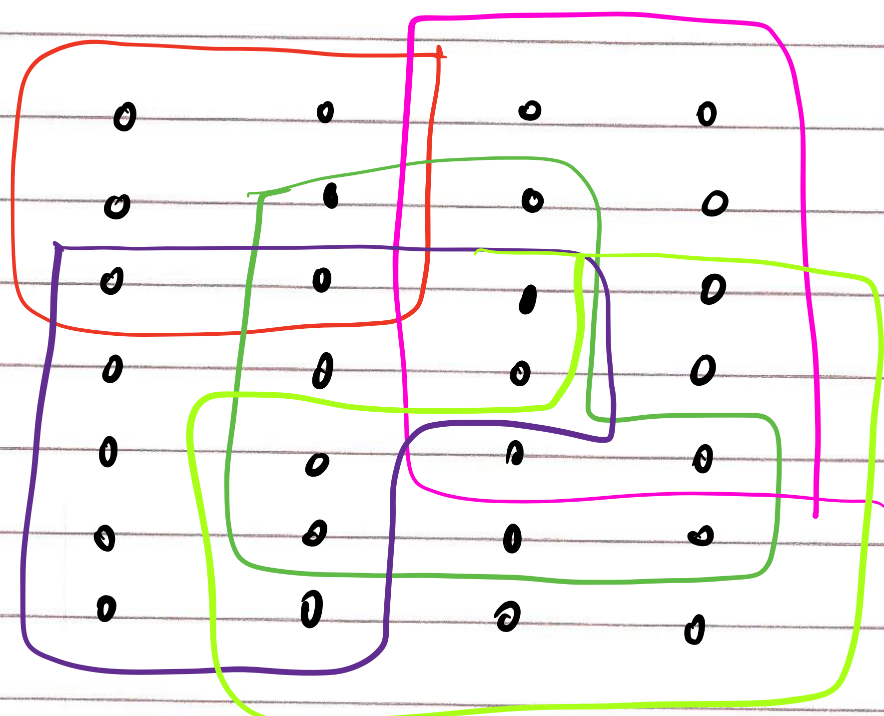
Proof: If we have a blackbox to solve vertex cover, we can decide if G has an independent set of size at least $\underline{k}$, by asking the blackbox if G has a vertex cover of size at most $\underline{n-k}$.

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 if G has an indep. set of size at least $\underline{n-k}$.

Set Cover Problem

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



Claim: Vertex Cover $\le_p$ Set Cover

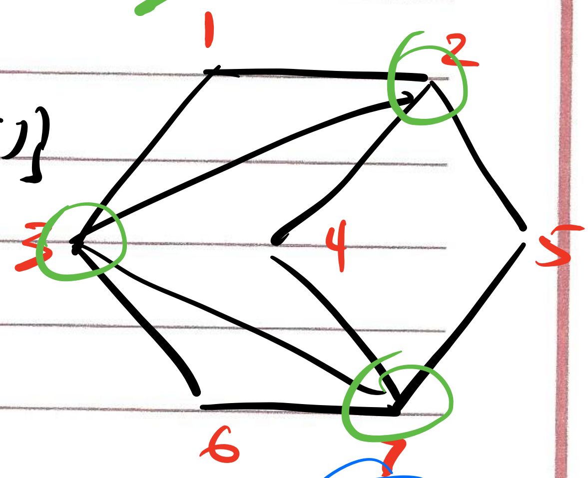
$S_1 = \{(1,2),(1,3)\}$

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

$S_3 =$

$\vdots$

$S_7 =$



k $\qquad$ G, k

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

Proof:

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

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

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

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

$(x_1 \vee \bar{x}_2)$ → $x_1 = 0, x_2 = 1$ ✗

$x_1 = 1, x_2 = 1$ ✓

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

- An assignment satisfies a collection of clauses if

$C_1 \wedge C_2 \wedge \ldots C_k$

~~it~~ evaluates to 1.

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

$x_1 = 1$, $x_2 = 1$, $x_3 = 1$ ✗

$x_1 = 0$, $x_2 = 0$, $x_3 = 0$ ✓

$\vdots$

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

Satisfiability Problem (SAT)

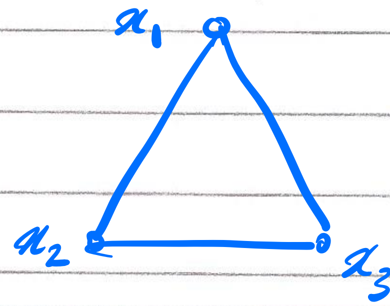
Problem statement: Given a set of clauses $C_1 ... C_k$ each of length 3 over a set of variables $X = \{x_1 .. x_n\}$ does there exist a satisfying truth assignment?

3SAT

Claim: 3SAT $\leq_p$ Independent Set

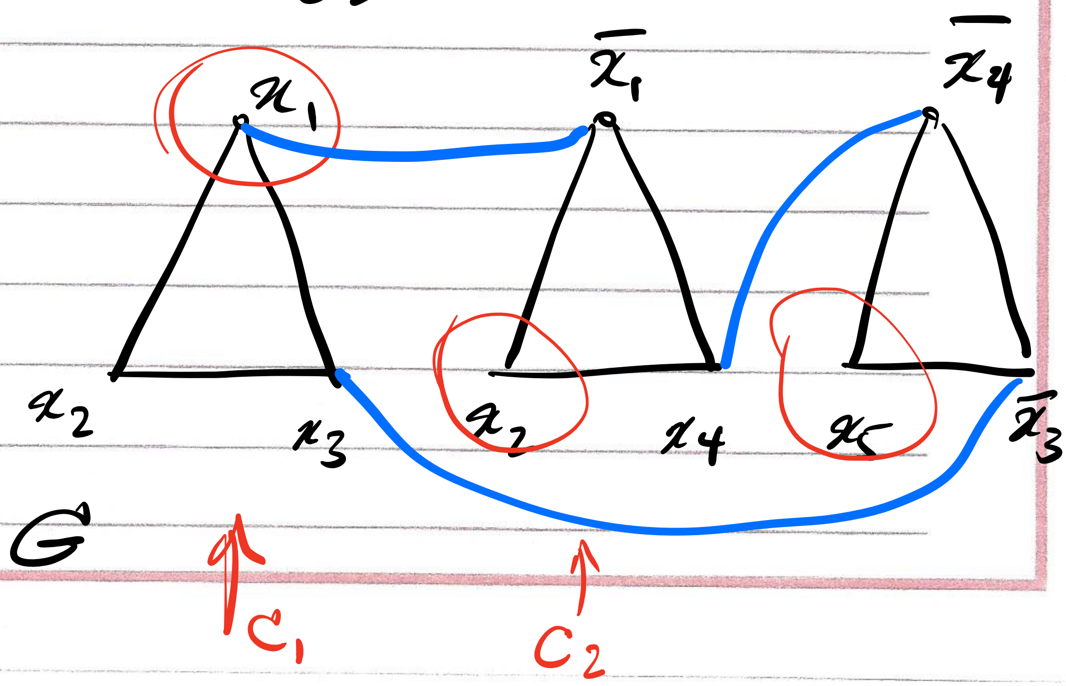
Plan: Given an instance of 3SAT with k clauses, build a graph G that has an indep. set of size k iff the 3SAT instance is satisfiable.

$(x_1 \vee x_2 \vee x_3)$



ex.

$$\left.\begin{aligned} C_1 &= (x_1 \vee x_2 \vee x_3) \\ C_2 &= (\bar{x}_1 \vee x_2 \vee x_4) \\ C_3 &= (\bar{x}_4, x_5 \vee \bar{x}_3) \end{aligned}\right\} k$$



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

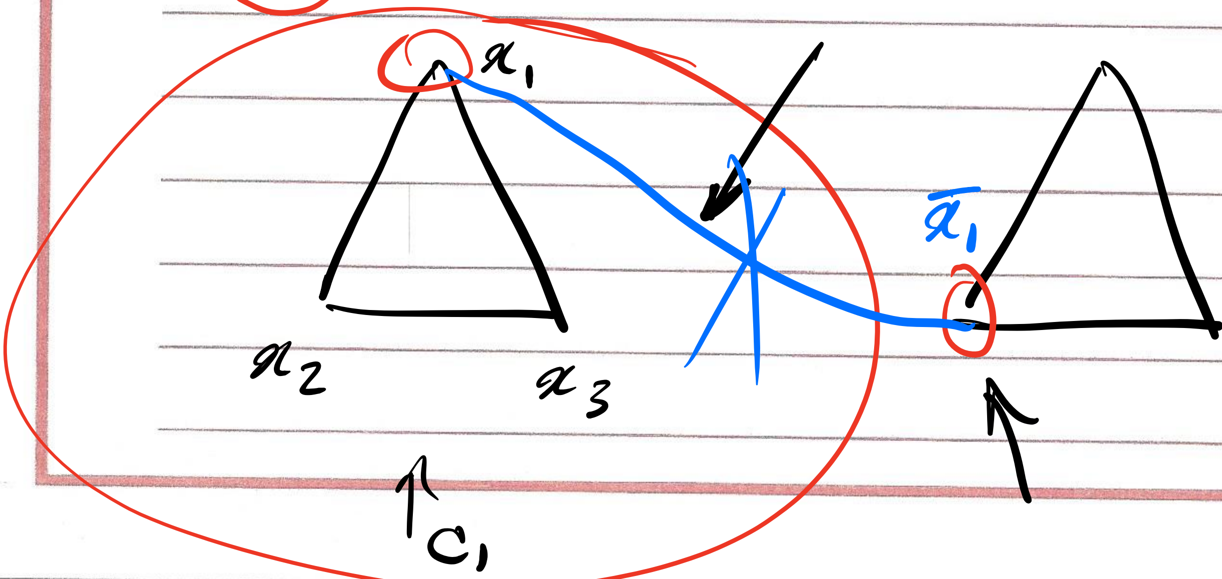
Proof: A) If the 3-SAT instance is satisfiable, then there is at least one node label per triangle that evaluates to 1.

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

$$\left.\begin{aligned} C_1 &= (x_1 \vee x_2 \vee x_3) \\ C_2 &= (\bar{x}_1 \vee x_2 \vee x_4) \\ C_3 &= (\bar{x}_4 \vee x_5 \vee \bar{x}_3) \end{aligned}\right\} k$$

$x_1 = 1,\ x_2 = 1,\ x_3 = 1$

$x_4 = 0,\ x_5 = 0$



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 to 1

if $\bar{x}_i$ appears as a label in S then set x_i to 0

if neither x_i nor $\bar{x}_i$ appear as a label in S, then set x_i to either 0 or 1

Efficient Certification

To show efficient certification:

1. Polynomial length certificate

2. Polynomial time certifier

Efficient certification

3-SAT

Certificate $\underline{t}$ is an assignment of truth values to variables (x_i)

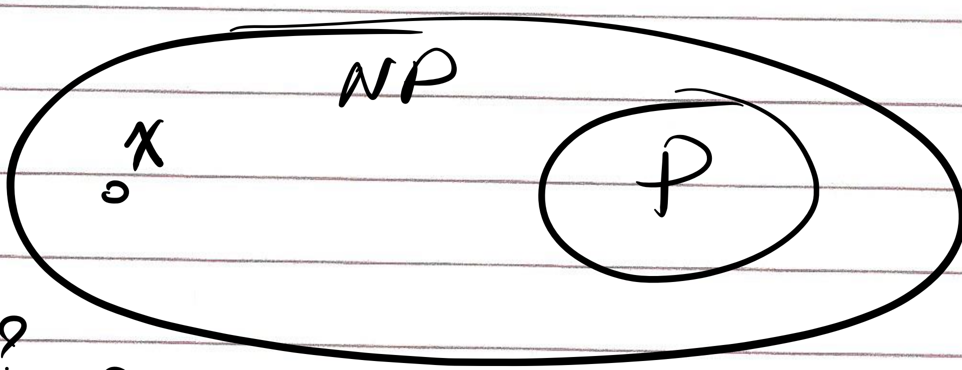
Certifier: evaluate the clauses. if all of them evaluate to $\underline{1}$ then it answers yes.

Indep Set

Certificate $\underline{t}$ is a set of nodes of size at least $\underline{k}$ 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



$P \stackrel{?}{=} NP$

if $X \in NP$ and for all $Y \in NP$

$Y \le_p \underline{X}$, then X is the hardest prob. in NP.

3SAT has been proven to be such a problem

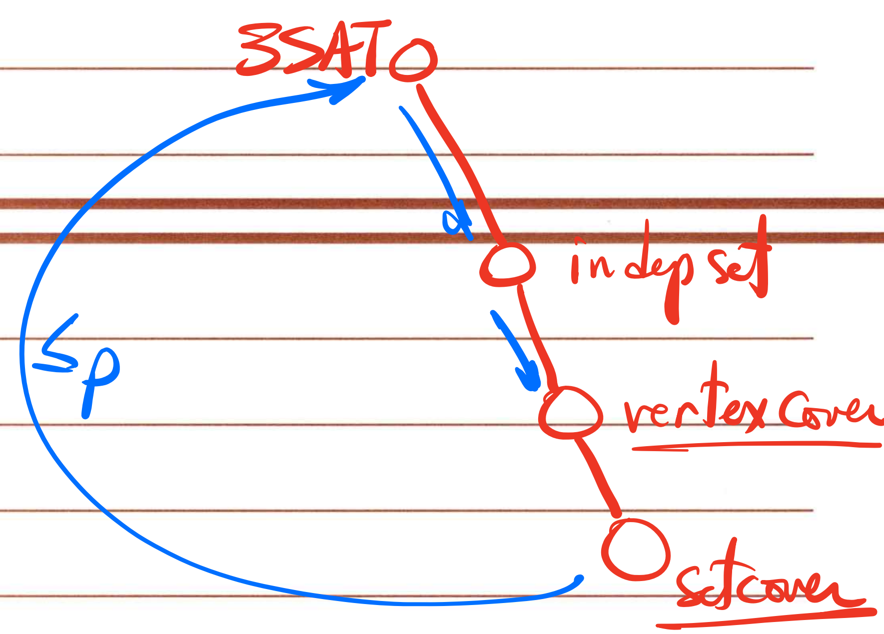
Such a problem is called NP-Complete.

Transitivity

if $Z \leq_p Y$ and $Y \leq_p X$

thus $Z \leq_p X$

3SAT $\leq_p$ indep set $\leq_p$ vertex cover $\leq_p$ setcover



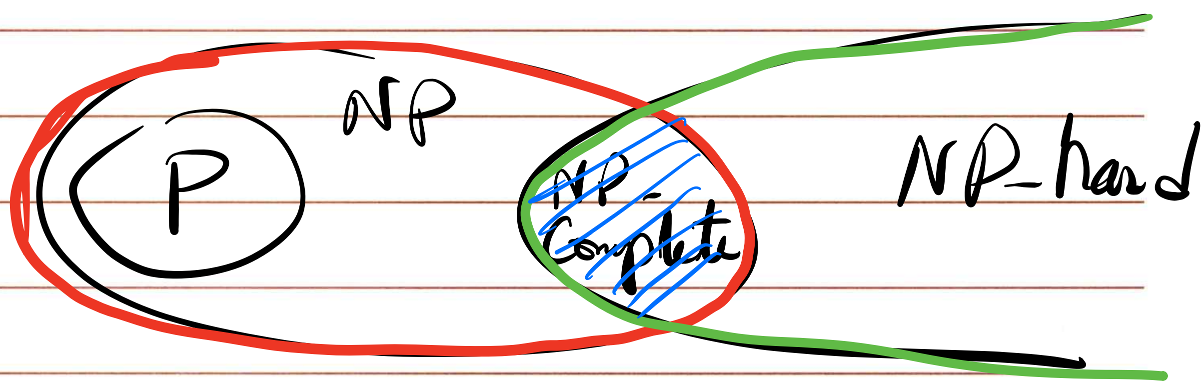
Basic strategy to prove
a prob. X is NP-complete

1- Prove $X \in NP$

2- Choose a problem Y that
is known to be NP-Complete

3- Prove that $Y \leq_p X$

NP-hard is the set of problems
that are at least as hard
as NP-Complete problems.



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. 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.)

SAT	3SAT
(x_1)	$(x_1 \vee x_1 \vee x_1)$
$(x_1 \vee x_2)$	$(x_1 \vee x_2 \vee x_1)$
$(x_1 \vee x_2 \vee x_3)$	$(x_1 \vee x_2 \vee x_3)$
$(x_1 \vee x_2 \vee x_3 \vee x_4)$	$(x_1 \vee x_2 \vee S_1) \wedge (\overline{S_1} \vee x_3 \vee x_4)$

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

$(x_1 \vee x_2 \vee S_1) \wedge$
$(\overline{S_1} \vee x_3 \vee S_2) \wedge$
$(\overline{S_2} \vee x_4 \vee x_5)$

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.

1- Show Set Packing is in NP

Certificate: List of k sets that have no elements in Common.

Certifier: check that the intersection between each pair of sets in the list is Null &

there are k sets in the Certificate

2- Choose Indep set.

3- Show indep set $\le_p$ Set packing

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

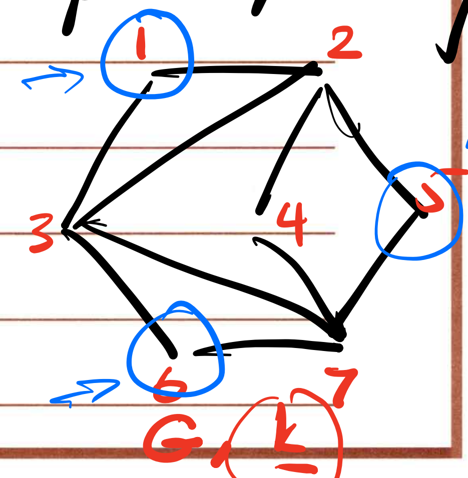
$S_2 = \{(1,2),(2,3),(2,4),(2,5)\}$

$S_3 = \vdots$

S_5

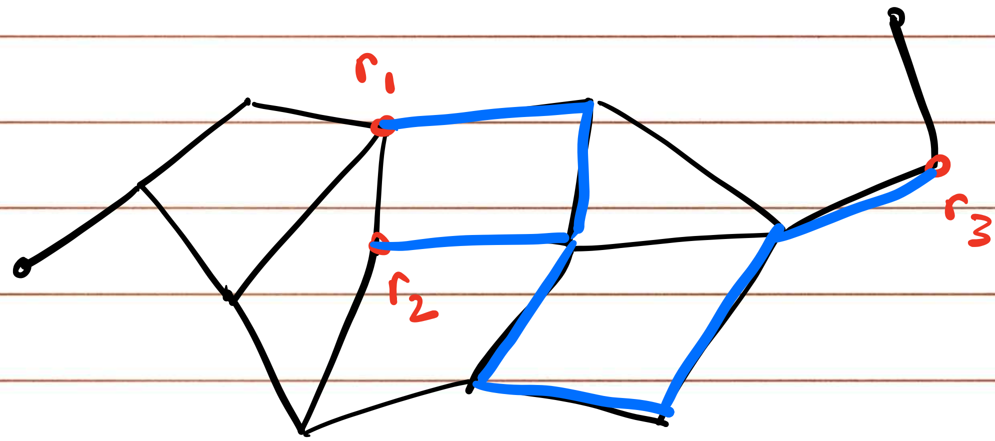
S_6

$S_7 = - - - -$



K

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- Show Steiner Tree $\in$ NP

a- Certificate : Tree spanning T across all nodes in set R (and maybe some in S) w/ Cost $\leq C$

b- Certifier :

- check that T is a tree
- " " cost of $T \leq C$
- " " T covers all nodes in R.

2- Choose vertex Cover

3- Show Vertex Cover $\leq_p$ Steiner Tree