HW10 out later today (due next Tue 8 pm)

HW11

SO FAR

Part 1 : MODELS OF COMPUTATION

Part 2 : DESIGN EFFICIENT ALGORITHMS

Part 3 : Problems for which NO ALGORITHMS exist

Problems for which NO EFFICIENT ALGORITHMS exist.

HOW TO ARGUE THAT SOLUTIONS ARE UNLIKELY TO EXIST?

TEMPLATE

- Suppose you're trying to figure out if there exists an efficient algorithm for problem Y.
  - You have a channel to God.
  - God will only tell you whether a DIFFERENT PROBLEM X is hard. (no solutions)

If Y has a solution, then so does X.
God told you X is hard.
$\Rightarrow$ Y is also hard.

CONDITIONAL RESULTS

DECISION PROBLEMS

PROBLEM. Π : collection of instances (strings).

for each instance, answer is YES or NO

Answer function $f_\pi : \Sigma^* \rightarrow \{0,1\}$ where

$f_\pi(I) = 1$ iff I is YES instance

$f_\pi(I) = 0$ iff I is NO instance

$$L_\pi = \{ I \mid f_\pi(I) = 1 \}$$

$\langle x \rangle$ refers to an encoding of x in some format

Graph G. $\langle G \rangle$ is an encoding of the graph as a string.

$\underline{G = (V, E)}$, s, t, B ← length of shortest path from s to t in G

Instance $= \langle G, s, t, B \rangle$.

REDUCTION BETWEEN LANGUAGES.

For two languages L_x, L_y

A reduction FROM L_x TO L_y is an algorithm:

input : $w \in \Sigma^*$

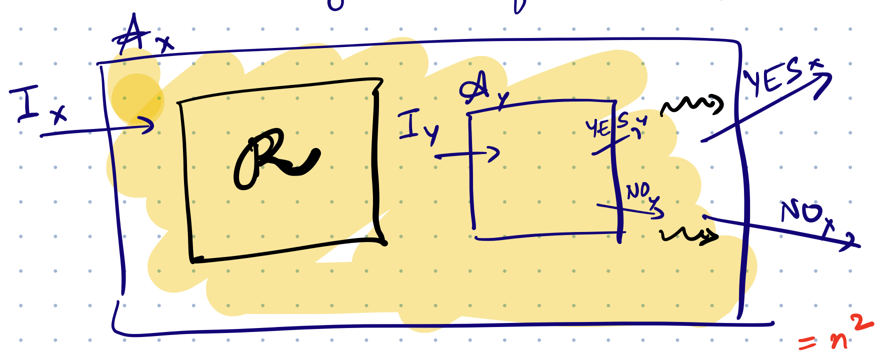
output : $w' \in \Sigma^*$

such that $w' \in L_y \iff w \in L_x$.

R: Reduction (from) $X \xrightarrow{(\text{to})} Y$ $\qquad X \leq Y$

Given A_y: Algorithm for Y.

Build A_x: Algorithm for X. (that uses A_y)



R has running time $\underbrace{R(n)}_{}^{= n^2}$ where n is size of input to R

A_y has running time $Q(n)$ $= n^{1.5}$ where n is size of input to A_y

A_x has running time? $\boxed{R(n) + Q(R(n))}$ $\quad n^2 + n^3 \qquad Q(R(n)) = n^3$

Suppose $|I_x| = n$. First run R, takes $R(n)$.

Next run A_y, which takes $Q(|I_y|) \leq Q(R(n))$

> If R is polynomial-time and A_y is also polynomial time, then A_x is polynomial time.

If R is polynomial time and makes polynomially many accesses to A_y, and A_y is also polynomial time, then A_x is polynomial time.

<u>Lemma</u> (1) If $X \leq Y$ and Y has an algorithm, then X has an algorithm.

(2) If $X \leq_p Y$ and Y has a polynomial-time algorithm, then X has a polynomial-time algorithm.

poly-time reduction

(3) If $X \leq Y$ and <u>X does not have an algorithm</u>

Y does not have an algorithm.

(4) If $X \leq_p Y$ and <u>X does not have a poly-time algorithm</u>

Y does not have a polynomial-time algorithm.

$$X \leq Y, \; Y \leq Z \Rightarrow X \leq Z$$

$$X \leq_{(P)} Y, \; Y_{(P)} \leq Z \Rightarrow X \leq_{(P)} Z$$

$$X \leq Y \not\Rightarrow Y \leq X$$

PROVE HARDNESS OF NEW PROBLEM Y,

BASED ON KNOWN HARDNESS OF

WELL-KNOWN PROBLEM X.

$X \leq Y$ $\qquad$ $Y \leq X$

You know: X is hard.

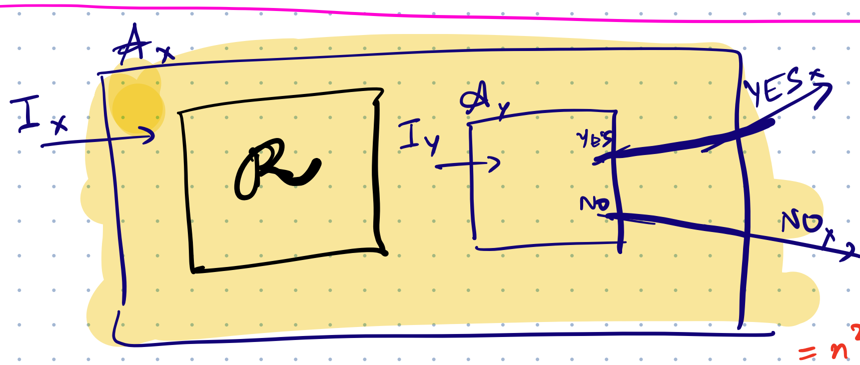
and $X \leq Y$

$\Rightarrow Y$ is hard.

How to prove $X \leq Y$.

Give $R(I_x) \rightarrow I_y$

$\boxed{\text{Such that } I_x \text{ is YES instance of } X \iff I_y \text{ is YES instance of } Y}$



$= n^2$

I_x is $YES_x \Rightarrow I_y$ is YES_y

I_y is $YES_y \Rightarrow I_x$ is YES_x

$\boxed{\text{How to prove } X \leq_p Y\ ?}$

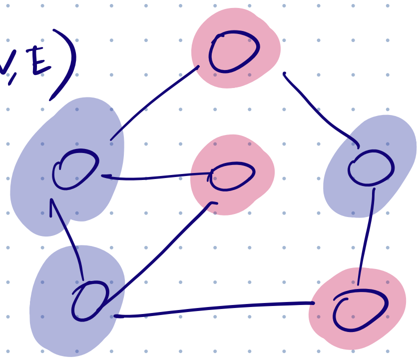
In addition to proving that

I_x is $YES_x \iff I_y$ is YES_y

also prove that R is polynomial-time.

EXAMPLES OF REDUCTIONS

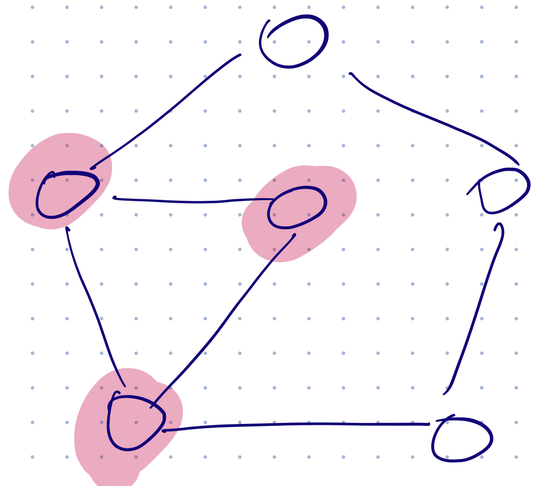
$G = (V, E)$



$\langle G, k \rangle$

Does G have an INDEPENDET SET of size $\geq k$?

INDEPENDENT SET: $S \subseteq V$ such that no 2 vertices in S are connected by an edge.

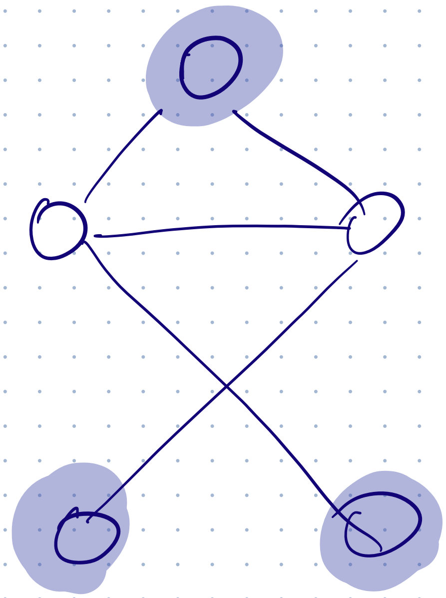


$\langle G, k \rangle$

G has a clique of size $\geq k$?

CLIQUE: Set $S \subseteq V$ s.t. every pair of vertices in S is connected by an edge

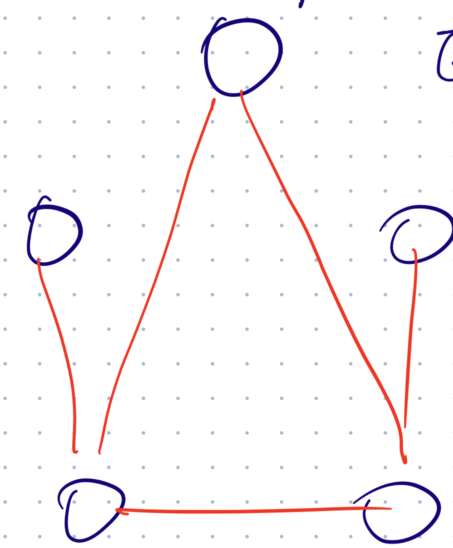
$G =$



(G, k)

Does G have ind. set of size $\geq k$?

R : given G, computes $\overline{G}$



$\overline{G}$ has edge $(u,v) \iff (u,v)$ is NOT an edge in G.

To prove: $I_x = (G,k)$ is a YES instance of INDEPENDENT SET $\iff I_y = (\overline{G}, k)$ is a YES instance of CLIQUE.

G has an independent set of size $\geq k$

$\Longleftrightarrow$ $\overline{G}$ has a clique of size $\geq k$.

A set S is an independent set in G

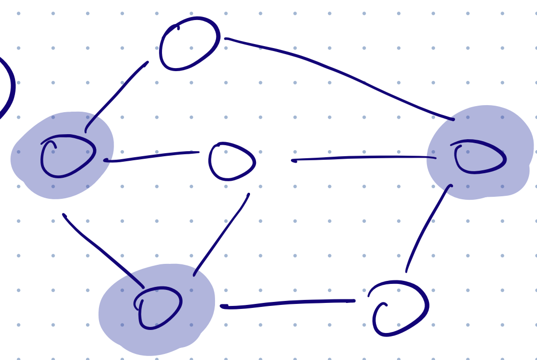
$\Longleftrightarrow$ no 2 vertices in S have an edge between them in G

$\Longleftrightarrow$ every pair of vertices in S have an edge between them in $\overline{G}$

$\Longleftrightarrow$ S is a clique in $\overline{G}$.

VERTEX COVER

$G=(V,E)$



Set of vertices S is V.C. if every edge $e \in E$ has at least 1 endpoint in S.

Let $G=(V,E)$ be a graph

$\boxed{S \text{ is an independent set} \Longleftrightarrow V \setminus S \text{ is a vertex cover.}}$

INDEPENDENT SET $\leq$ VERTEX COVER

Consider any $uv \in E$

$u \notin S$ or $v \notin S$ (S is indep set)

$\Rightarrow u \in V \setminus S$ or $v \in V \setminus S$

$\Rightarrow V \setminus S$ is a vertex cover.

$V \setminus S$ is V.C.

Consider any $u, v \in S$.

$\Rightarrow uv$ is not an edge of G else $V \setminus S$ does not cover uv

$\Rightarrow S$ is an independent set.

$(G, k) \in_{YES}$ INDSET $\iff$ $(G, n-k) \in_{YES}$ VCOV.

INDSET $\leq_p$ VCOV

VCOV $\leq_p$ INDSET