CS 401

P and NP

Xiaorui Sun

Stuff

Homework 4 due tomorrow

Homework 5 will be released later today (due May 6)

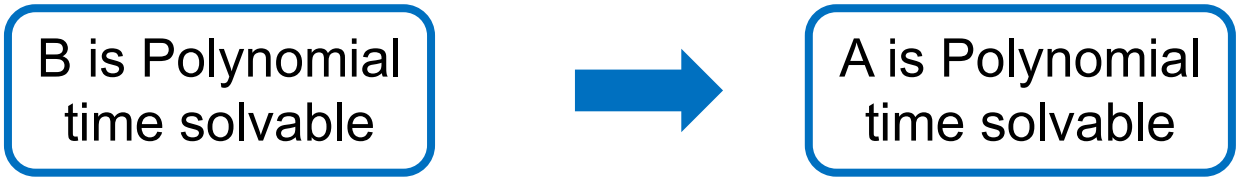
- Helpful for final exam preparation

Polynomial Time Reduction

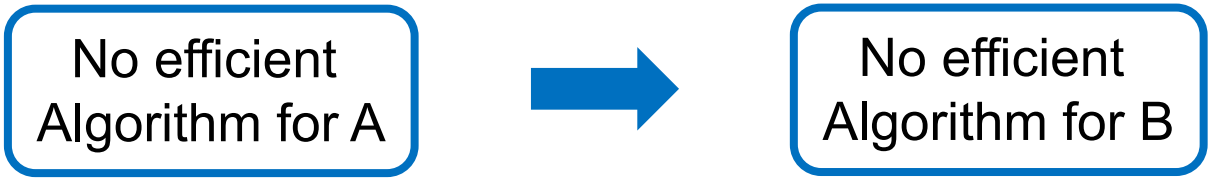
Def A $\leq_P$ B: if there is an algorithm for problem A using a 'black box' (subroutine) that solve problem B s.t.,

- Algorithm uses only a polynomial number of steps
- Makes only a polynomial number of calls to a subroutine for **B**

So,



Conversely,



In words,

- Problem A is polynomial-time reducible to problem B
- B is as hard as A (it can be even harder)
- Informally, A is a special case of B

Polynomial Time Reduction

Basic reduction strategies

- Reduction by simple equivalence.
- Reduction from special case to general case.
- Reduction by encoding with gadgets.

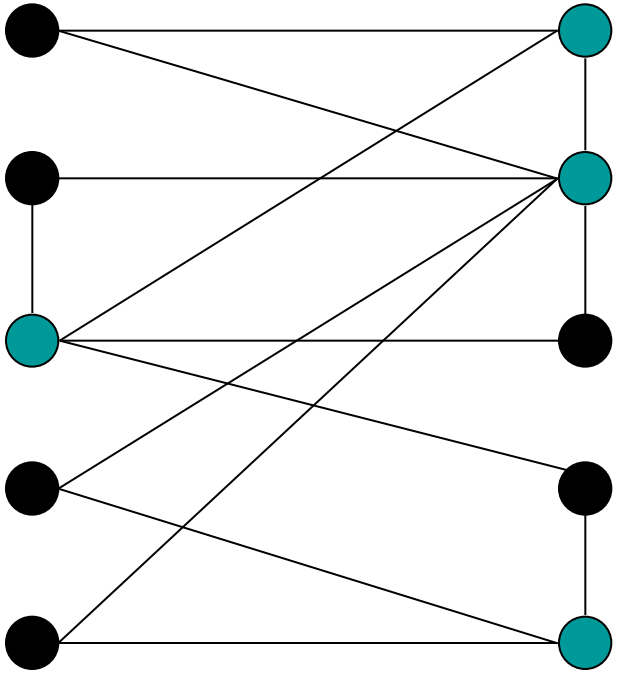
More advanced technique, read KT 8.2

Example 1: Vertex Cover $\equiv_p$ Indep Set

VERTEX COVER: Given a graph G = (V, E) and an integer k, is there a subset of vertices $S \subseteq V$ such that $|S| \leq k$, and for each edge, at least one of its endpoints is in S?

Ex. Is there a vertex cover of size ≤ 4? Yes.

Ex. Is there a vertex cover of size ≤ 3? No.



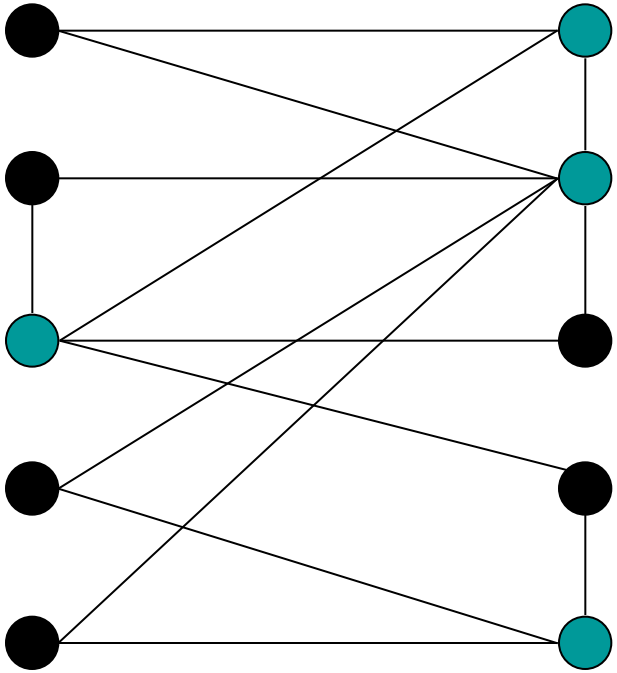
vertex cover

Example 1: Vertex Cover $\equiv_p$ Indep Set

INDEPENDENT SET: Given a graph G = (V, E) and an integer k, is there a subset of vertices $S \subseteq V$ such that $|S| \geq k$, and for each edge at most one of its endpoints is in S?

Ex. Is there an independent set of size ≥ 6? Yes.

Ex. Is there an independent set of size ≥ 7? No.



independent set

6

Example 1: Vertex Cover $\equiv_p$ Indep Set

Claim: For any graph $G = (V, E)$, S is an independent set iff $V - S$ is a vertex cover

Pf: =>

Let S be a independent set of G

Then, S has at most one endpoint of every edge of G

So, $V - S$ has at least one endpoint of every edge of G

So, $V - S$ is a vertex cover.

<= Suppose $V - S$ is a vertex cover

Then, there is no edge between vertices of S (otherwise, $V - S$ is not a vertex cover)

So, S is an independent set.

To show Vertex Cover $\leq_P$ Indep Set:
Algorithm for Vertex Cover(G, k)

- Run Indep Set(G, n - k)
- Return the solution of Step 1

Polynomial Time Reduction

Basic reduction strategies

- Reduction by simple equivalence.
- Reduction from special case to general case.

Example 2: Vertex Cover $\leq_p$ Set Cover

SET COVER: Given a set U of elements, a collection $S_1, S_2, \ldots, S_m$ of subsets of U, and an integer k, does there exist a collection of $\leq$ k of these sets whose union is equal to U?

Ex:

U = { 1, 2, 3, 4, 5, 6, 7 }

k = 2

S_1 = {3, 7}

S_2 = {3, 4, 5, 6}

S_3 = {1}

S_4 = {2, 4}

S_5 = {5}

S_6 = {1, 2, 6, 7}

Example 2: Vertex Cover $\leq_p$ Set Cover

Claim: VERTEX-COVER $\leq_P$ SET-COVER.

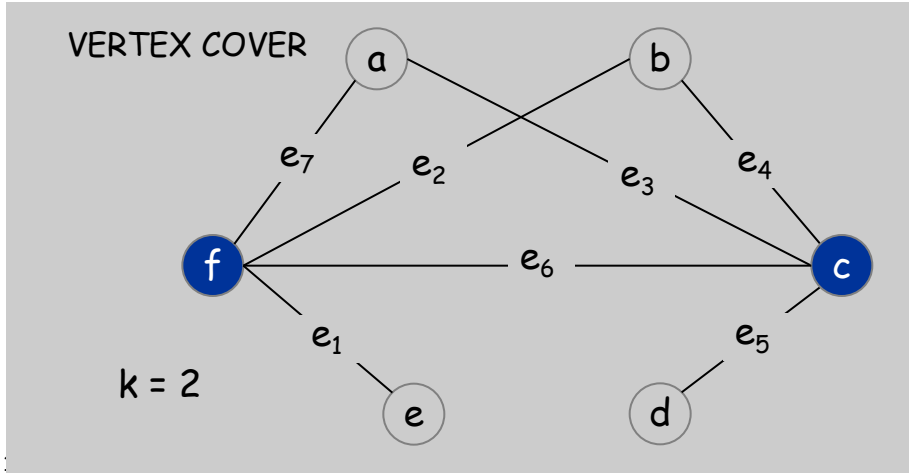
Pf: Given a VERTEX-COVER instance G = (V, E), k, we construct a set cover instance whose size equals the size of the vertex cover instance.

Construction:

Create SET-COVER instance:

- k = k, U = E, S_v = {e $\in$ E : e incident to v }

Set-cover of size $\leq$ k iff vertex cover of size $\leq$ k. ▪



SET COVER

U = { 1, 2, 3, 4, 5, 6, 7 }
k = 2
S_a = {3, 7}
S_b = {2, 4}
S_c = {3, 4, 5, 6}
S_d = {5}
S_e = {1}
S_f = {1, 2, 6, 7}

Review

Basic reduction strategies:

- Simple equivalence: INDEPENDENT-SET $\equiv_P$ VERTEX-COVER.
- Special case to general case: VERTEX-COVER $\leq_P$ SET-COVER.

Transitivity. If $X \leq_P Y$ and $Y \leq_P Z$, then $X \leq_P Z$.

Ex: INDEPENDENT-SET $\leq_P$ VERTEX-COVER $\leq_P$ SET-COVER.

P and NP

Decision Problems

Decision problem

- X is a set of strings.
- Instance: string s.
- Algorithm A solves problem X: A(s) = `yes` iff $s \in X$.

Polynomial time Algorithm A runs in poly-time if for every string s, A(s) terminates in at most p(|s|) "steps", where p(·) is some polynomial.

↑
length of s

PRIMES: X = { 2, 3, 5, 7, 11, 13, 17, 23, 29, 31, 37, …. }

Algorithm [Agrawal-Kayal-Saxena, 2002] $p(|s|) = |s|^8$.

Definition of P

P: Decision problems for which there is a poly-time algorithm.

Problem	Description	Algorithm	Yes	No
MULTIPLE	Is x a multiple of y?	Long division	51, 17	51, 16
RELPRIME	Are x and y relatively prime?	Euclid (300 BCE)	34, 39	34, 51
PRIMES	Is x prime?	AKS (2002)	53	51
EDIT-DISTANCE	Is the edit distance between x and y less than 5?	Dynamic programming	niether neither	acgggt ttttta
LSOLVE	Is there a vector x that satisfies Ax = b?	Gauss-Edmonds elimination	$\begin{bmatrix} 0 & 1 & 1 \\ 2 & 4 & -2 \\ 0 & 3 & 15 \end{bmatrix}, \begin{bmatrix} 4 \\ 2 \\ 36 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

NP

Certification algorithm intuition

- Certifier views things from "managerial" viewpoint.
- Certifier doesn't determine whether $s \in X$ on its own; rather, it checks a proposed proof t that $s \in X$.

Def Algorithm C(s, t) is a certifier for problem X if for every string s, $s \in X$ iff there exists a string t such that C(s, t) = `yes`.

"certificate" or "witness"

NP Decision problems for which there exists a poly-time certifier.

C(s, t) is a poly-time algorithm and
$|t| \leq p(|s|)$ for some polynomial $p(\cdot)$.

Remark NP stands for nondeterministic polynomial-time.