Design and Analysis of Algorithms

Algorithms-II : CS345A

Website: hello.iitk.ac.in

Lecture 29

NP Completeness – I

Optimization problems

Optimization problem:

Compute *<u>maximum</u>* *flow from s to t in graph G*

How can you use algorithm for the **optimization** version of a problem to solve its **Decision** version ?

Decision Problem:

Does there exist a flow

Yes-instances:

No-instances:

Try at least for the above example ?

POLYNOMIAL TIME REDUCTION

$$A \leq_P B$$

$A \leq_P B$

Definition:

Problem A is said to be

if there exists

Input:

Output:

- For each instance I of A :

  I is a yes-instance of A <u>if and only if</u> $f(I)$ is a yes-instance of B

- f is a polynomial time algorithm.

$A \le_P B$

Algorithmic Consequence	**Complexity theoretic consequence**
• We can use an algorithm for B to solve problem A.	No polynomial time algorithm for A ➔ No polynomial time algorithm for B.
• If B is polynomial time solvable, ➔ A is also polynomial time solvable	"B is computationally **at least** as hard as A"

EXAMPLE 1

Vertex Cover and Independent Set

Vertex Cover

Definition: Given an undirected graph $G = (V, E)$,

a subset $X \subseteq V$

For each edge $(u, v) \in E$,

either $u \in X$ or $v \in X$

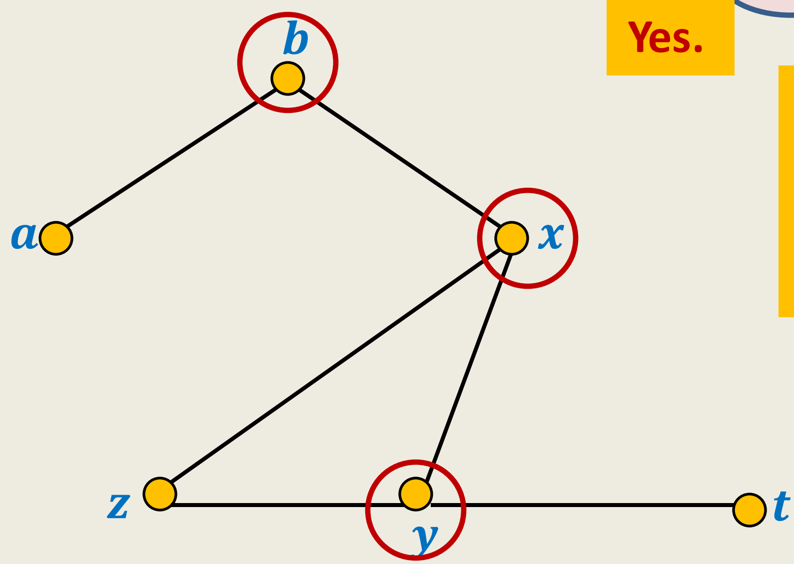
Is it a vertex cover now ?

Yes.

NO.

Reason:

None of (y, z) or (y, t) is covered



Optimization version:

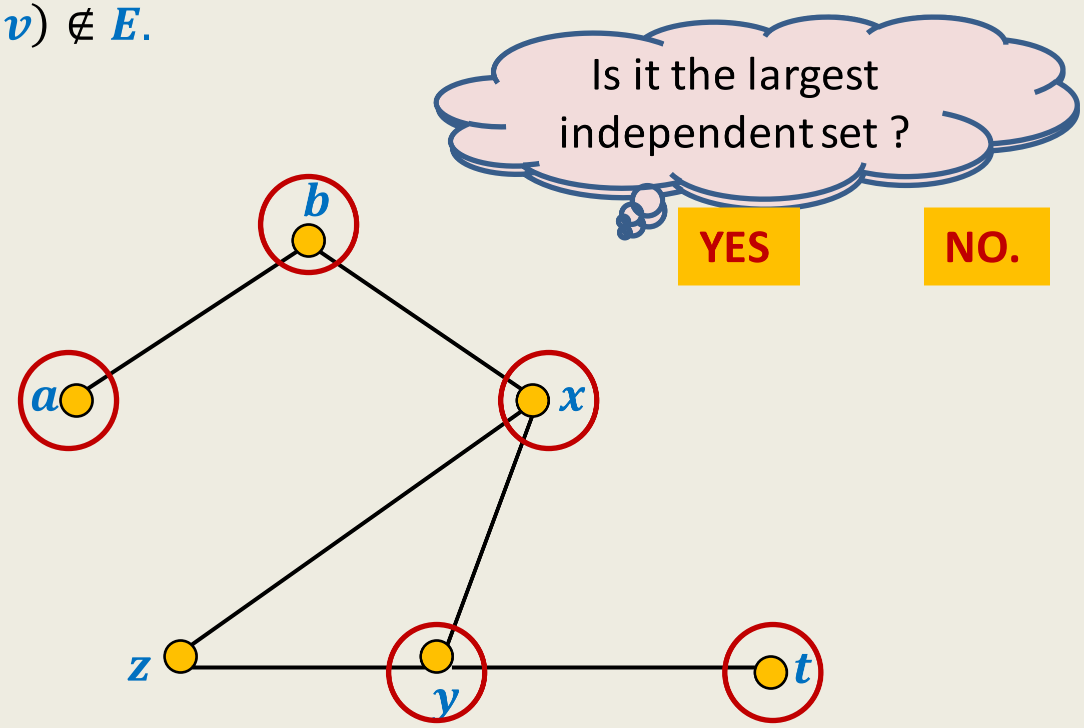
Decision version:

Independent Set

Definition: Given an undirected graph $G = (V, E)$,

a subset $X \subseteq V$ is said to be an **independent** set if

For each $u, v \in X$, $(u, v) \notin E$.



Optimization version:

Decision version:

VC $\leq_P$ IS

VC: Vertex Cover

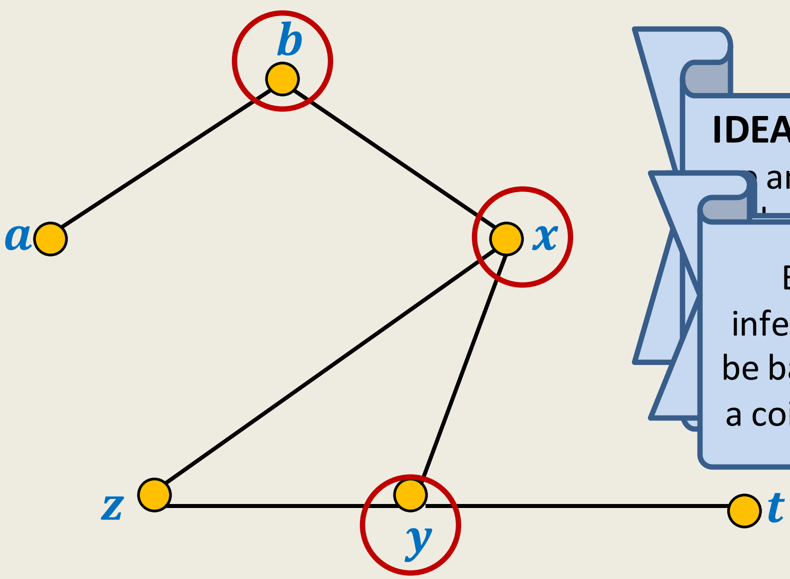
Input: an graph $G = (V, E)$ and $k \in Z^+$

Problem: Does there exist a vertex cover of size k?

IS: Independent Set

Input: an graph $G = (V, E)$ and $t \in Z^+$

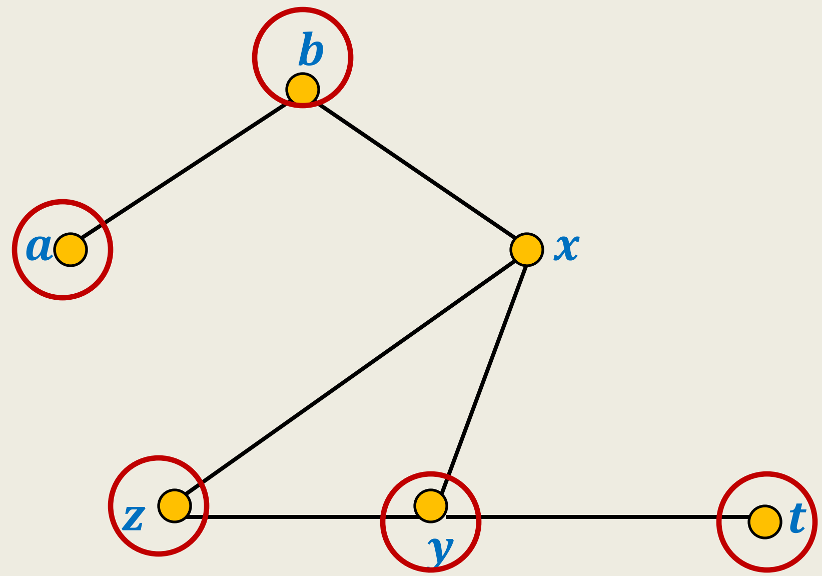
Problem: Does there exist an independent set of size t?



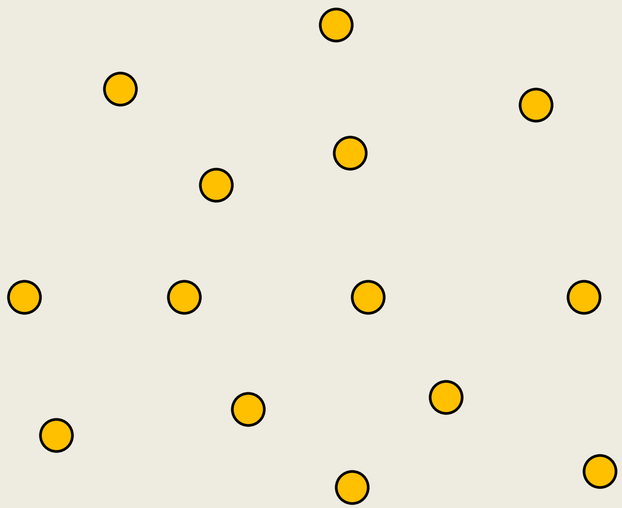
IDEA: Can you

any relatio

But your inference could be based on just a coincidence ☹

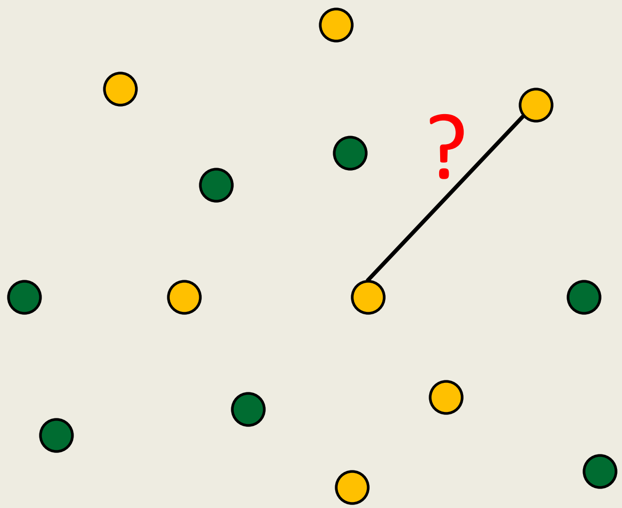


VC: **Vertex Cover**



IS: **Independent Set**

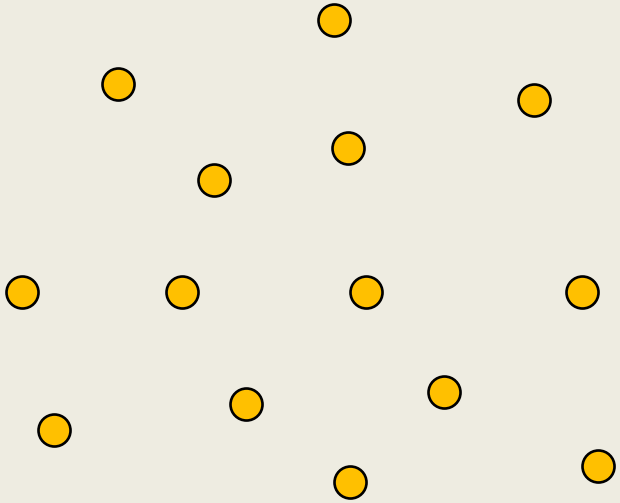
VC: Vertex Cover



IS: Independent Set

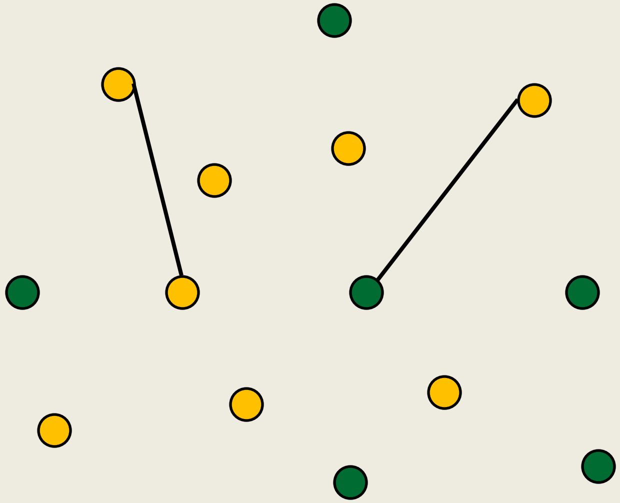
VC: Vertex Cover

IS: Independent Set



VC: Vertex Cover

IS: Independent Set



VC $\leq_P$ IS

Theorem: $\boldsymbol{X \subseteq V}$ is a vertex cover of $\boldsymbol{G}$ **if and only if** $\boldsymbol{V \setminus X}$ is an independent set of $\boldsymbol{G}$.

Proof:

- ➔
- 🡐

VC $\leq_P$ IS

Theorem (➔):

If $X \subseteq V$ is a vertex cover of G

Proof:

Let $Y = V \setminus X$

Consider any two vertices $u, v \in Y$.

Is it possible that $(u, v) \in E$?

No.

Reason:

$u \notin X$ and $v \notin X$

so if $(u, v) \in E$, then X is not a vertex cover.

Hence for each $u, v \in Y$, $(u, v) \notin E$.

➔ Y is an independent set of G

VC $\leq_P$ IS

Theorem (🡐):

If X is an independent set of G, then $V \backslash X$ is a vertex cover of G**.**

Proof:

Let $Y = V \backslash X$.

Consider any edge $(u, v) \in E$

Since X is an independent set,

So **at least** one of $\{u, v\}$ must be in Y.

So edge (u, v) is covered by Y.

🡒 Y is a vertex cover of G

VC $\leq_P$ IS

Thus the following theorem holds true.

Theorem: $X \subseteq V$ is a vertex cover of G **if and only if** $V \backslash X$ is an independent set of G.

Answer: On input (G, k), f outputs $(G, n-k)$.

Clearly f takes polynomial time.

Hence VC $\leq_P$ IS.

How to establish VC $\leq_P$ IS ?

Homework: Prove that IS $\leq_P$ VC

EXAMPLE 2

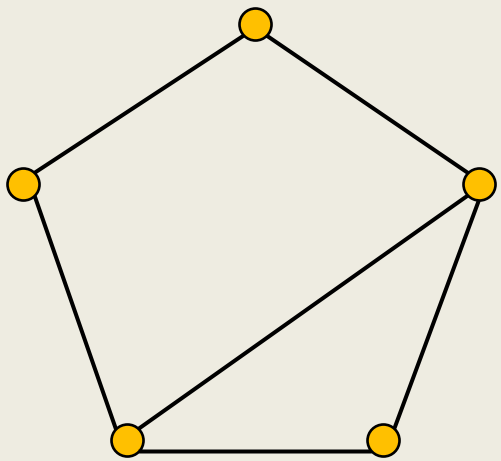
Hamiltonian cycle Problem

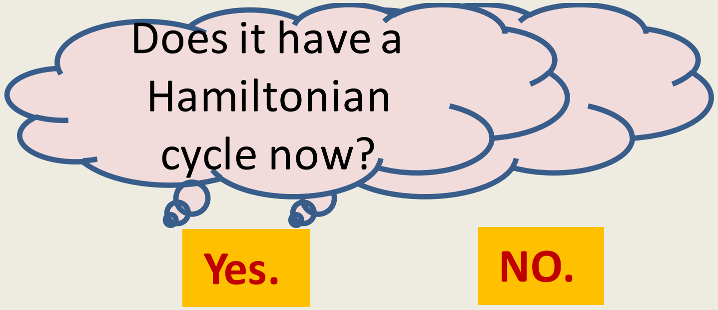
and

Traveling Sales person Problem

Hamiltonian Cycle Problem

Definition: Given an undirected graph $G = (V, E)$,
a cycle is said to be Hamiltonian





Optimization version:

Decision version:

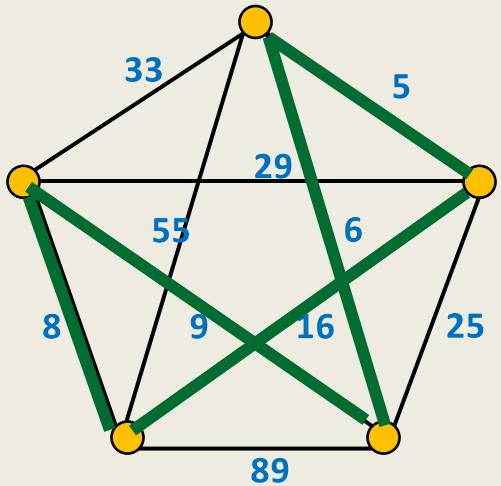
Traveling Sales Person Problem

Definition: In an undirected complete graph $G = (V, E)$ with nonnegative edge-cost, a tour is a sequence of vertices such that

- It originates and terminates at the same vertex
- There is an edge between every consecutive pair of vertices in the sequence
- Each vertex is visited exactly once.

Cost of tour: sum of cost of edges traversed in the tour.

This is the minimum cost TSP tour for this graph.



Optimization version:

Decision version:

HC $\leq_P$ TSP

HC: Hamiltonian Cycle

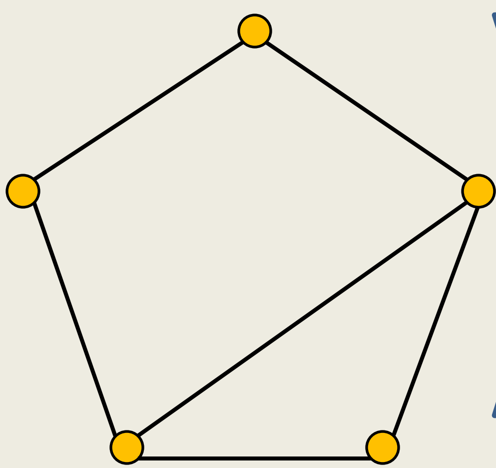
Input: An undirected graph

Problem: Does there exist a cycle passing through all vertices ?

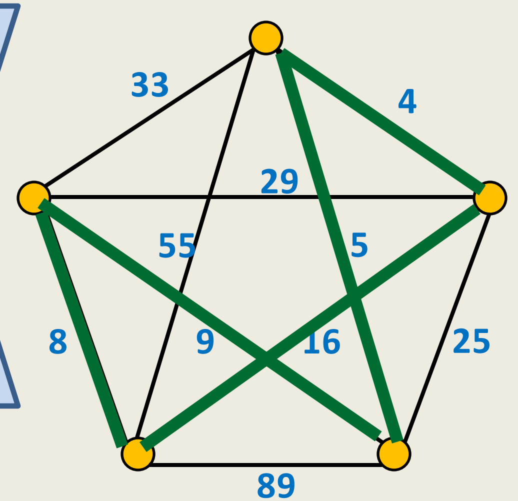
TSP: Traveling Sales Person

Input: An undirected <u>complete</u> graph with non-negative **cost** on edges and $\boldsymbol{b} \in \boldsymbol{R}^{+}$.

Problem: Does there exist a tour of cost $\boldsymbol{b}$?



IDEA: Build a weighted complete graph $\boldsymbol{G'}$ s.t. the Hamiltonian cycle in $\boldsymbol{G}$ turns out to be the least cost tour in $\boldsymbol{G'}$.



HC $\leq_P$ TSP

Let $G = (V, E)$ be an instance of **HC**

Build a complete graph G' on V vertices.

For each pair (u, v)

If $(u, v) \in E$, $c(u, v) \leftarrow$ **1**

else $c(u, v) \leftarrow$ **2**

(Any cost greater than **1** will work here)

Theorem: G has a **Hamiltonian cycle** if and only if G' has a **TSP** tour of cost at most **n**.

Proof:

- ➔
- 🡐

HC $\leq_P$ TSP

Theorem (➔):

If G has a **Hamiltonian cycle** then G' has a **TSP** tour of cost at most **n**.

Proof:

Let C be a **Hamiltonian cycle** in G.

G is a subgraph of G'.

So C must be present in G' as well.

C is a tour since each vertex appears exactly once in C.

Cost of each edge of C is **1** since each edge of C is present in G as well.

So the cost of tour C = **n**

Hence G' has a tour of cost at most **n**.

HC $\leq_P$ TSP

Theorem (🡐):

If G' has a **TSP** tour of cost at most **n**, then G has a **Hamiltonian cycle**.

Proof:

Let C be a **TSP** tour of cost at most **n** in G'.

Cost of each edge in G' is <u>at least</u> **1**.

There are **n** edges in C.

So each edge of C must have weight exactly 1.

Therefore, each edge of C is present in G as well.

So C is present in G as well.

Since each vertex appears exactly once on C, so C is **Hamiltonian** as well.

Hence G has a Hamiltonian cycle.

HC $\leq_P$ TSP

Thus the following theorem holds true.

Theorem: G has a **Hamiltonian cycle** if and only if G' has a **TSP** tour of cost at most **n**.

How to establish HC $\leq_P$ TSP ?

Answer: On input (G), f constructs G' as described above and outputs (G', n).

Clearly f takes polynomial time.

Hence HC $\leq_P$ TSP

Homework: A path in undirected graph is said to be **Hamiltonian** path if it passes through each vertex exactly once.

HP - **Hamiltonian** path problem :

“Does a given undirected graph G have a Hamiltonian path ?”

Prove that HC $\leq_P$ HP

$A \leq_P B$

Complexity theoretic consequence of $A \leq_P B$:

If there does not exist any polynomial time algorithm for A, then

There can not exist any polynomial time algorithm for B.

"B is computationally **at least** as hard as A"

NP
A CLASS OF PROBLEMS

The story behind its invention…

Go back to 1960's

Efficient algorithm was found.	No Efficient algorithm could be designed till date
Shortest Path	Longest Path
Minimum spanning Tree	Travelling salesman Problem
Euler tour	Hamiltonian cycle
Min Cut	Balanced Cut
Independent Set on trees	Independent Set
Bipartite matching	3D matching
Linear Programming	Integer Linear Programming
⋮	⋮

This motivated researchers to search for any common traits among all these problems.

It was quite surprising and even frustrating to be unable to find efficient algorithm for so many problems when their similar looking versions had very efficient algorithms.

- **Travelling Salesman Problem**

Decision version: Given a graph $\boldsymbol{G}$, does there exist a tour of cost at most $\boldsymbol{b}$?

Searching for a tour of cost at most $\boldsymbol{b}$ appears to be difficult ☹

But what about **verifying** whether a given sequence of vertices is a tour of cost at most $\boldsymbol{b}$?

It is quite easy ☺.

- **Vertex cover**

Decision version: Given a graph $\boldsymbol{G}$, does there exist a vertex cover of size $\leq \boldsymbol{k}$.

Searching for a subset of $\boldsymbol{k}$ vertices that is a vertex cover of $\boldsymbol{G}$ appears difficult ☹.

But what about **verifying** whether a given subset of $\boldsymbol{k}$ vertices is a vertex cover ?

It is quite easy ☺.

What about verifying a proposed solution ?

Easy

No Efficient algorithm till date ☹	
Longest Path	Is there a path of length $\geq k$ in G ?
Vertex cover	Does there exist a vertex cover of size $\leq k$ in G?
Travelling salesman Problem	Does there exist a tour of cost $\leq c$ in G?
Hamiltonian cycle	Does there exist a cycle of length n in G?
Independent Set	Does there exist an independent set of size $\geq k$ in G?
3D matching	⋮
Integer Linear Programming	⋮
⋮	

What if the answer of an instance is **Yes** ?

There is a short *certificate*.

No Efficient algorithm till date ☹	
Longest Path	short certificate Search: difficult verification: easy
Vertex cover	
Travelling salesman Problem	
Hamiltonian cycle	
Independent Set	
3D matching	
Integer Linear Programming	
⋮	

Homework: Try to formalize these 2 traits of these problems.