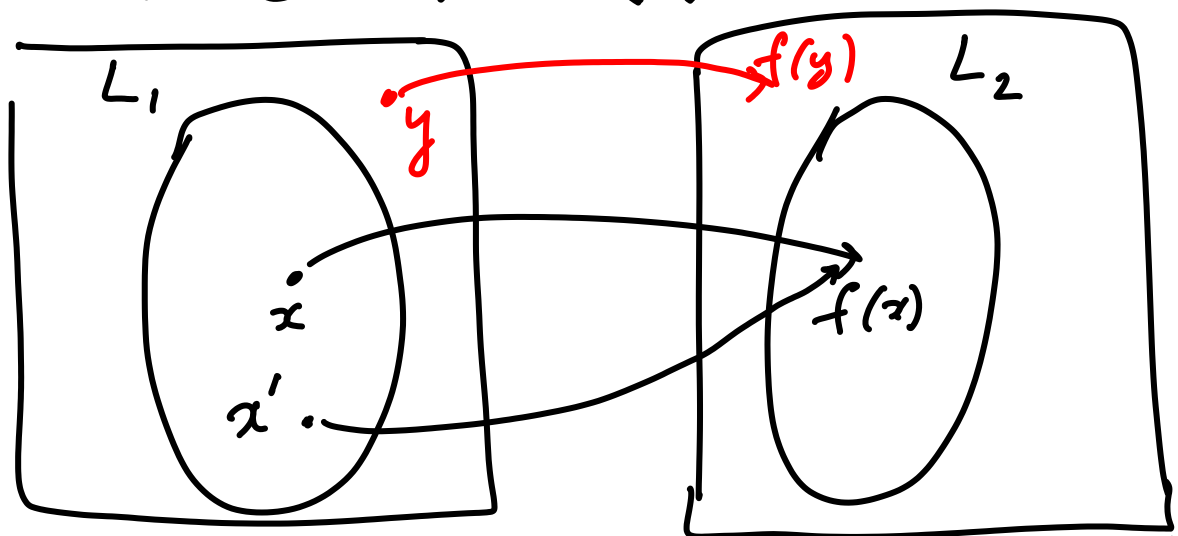
COL 702 Lecture 26 Nov 9

L_1 and L_2 are languages over some alphabet Σ.

L_1 is reducible to L_2 $\quad L_1 \leq L_2$

if there exists a function $f : \Sigma^* \to \Sigma^*$ s.t. $\forall x \in \Sigma^*$

$$x \in L_1 \iff f(x) \in L_2$$



Note: f need not be 1-1

If $f(x)$ can be computed in polynomial time then L_1 is polynomial-time reducible to L_2

<u>Claim</u>: If $L_1 \leq_{\text{polytime}} L_2$ and there is a polynomial-time algorithm to recognise L_2 then L_1 can also be recognised in polytime

The algorithm to recognize L_1

① Given any string x, we first compute $y = f(x)$ in polynomial time ($\text{polynomial in } |x|$: length of string)

$|y|$ is $\underbrace{\text{polynomial bounded}}_{p_1}$ by $|x|$

② We use y as an input to the algorithm for L_2 and the running time is polynomial in $|y|$

p_2

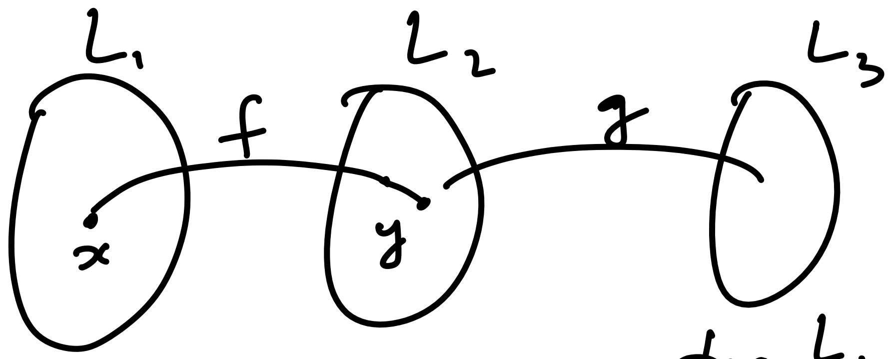
If $y \in L_2$ then return YES
else return NO

What is the running time

$p_2(p_1(|x|))$ which is a polynomial function

Claim: If $L_1 \leq_{poly} L_2$ and $L_2 \leq_{poly} L_3$

$\Rightarrow L_1 \leq_{poly} L_3$



$g(f)$ is the mapping for L_1 to L_3

$g(f(x))$

P: class of polynomial-time recognisable Languages

NP: class of languages recognisable in <u>non deterministic</u> polynomial time

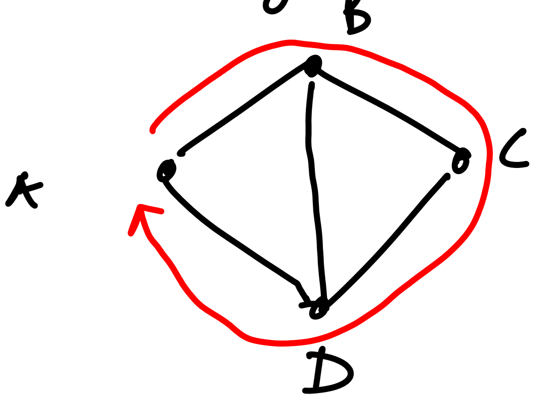
(It is not the same as languages not in polynomial time)

$P \subseteq NP$

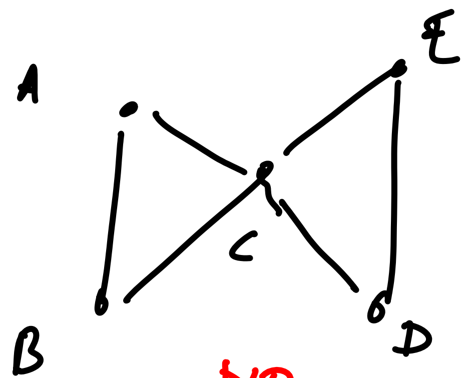
$$\boxed{P = NP \text{ or } P \subset NP}$$

Hamilton cycle problem

Given a graph $G = (V, E)$, is there a cycle that visits all vertices exactly once



YES



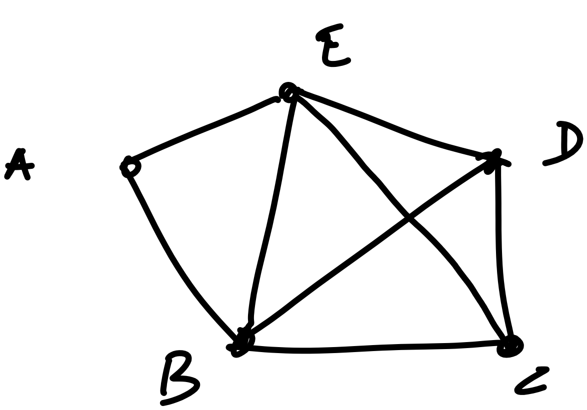
NO

Check all possible permutations for a legal tour

Independent Set Problem

Given a graph $G = (V, E)$ and int k

Is there an independent set of size $\geq K$



$K = 2$	YES
$K = 3$	NO
$K = 4$	NO

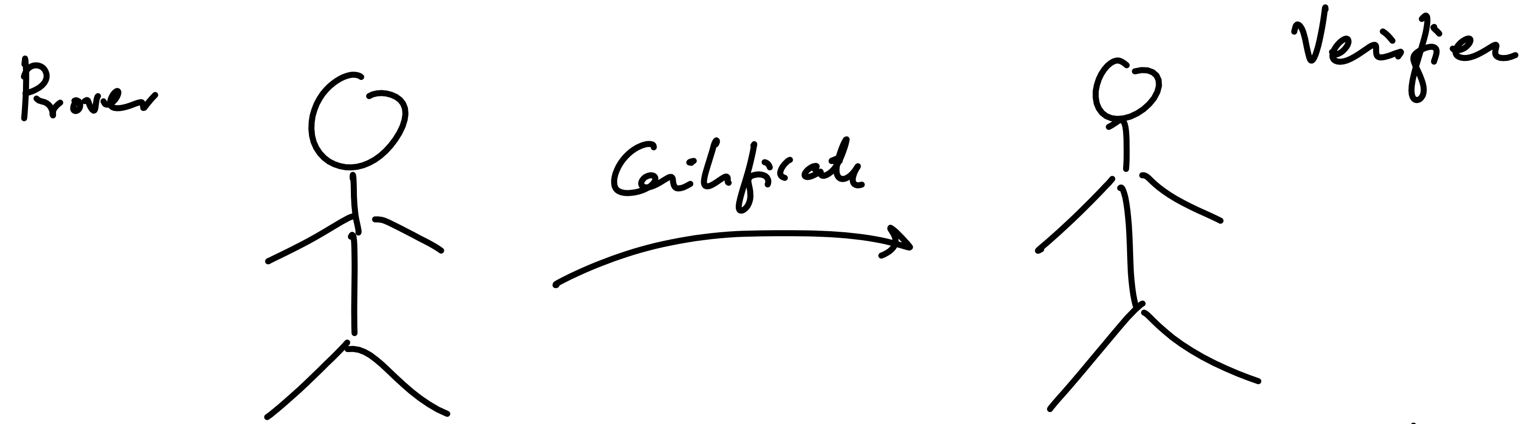
Try all possible subsets of size K and check independence

Guessing a tour

Start from vertex 1 → 2, 3, ⋮, n

guess the next vertex
guess . . .

G(V, E)



There is a polynomial time verification for the property of hamilton cycle

Non- Hamilton

No independent set of size k

Knapsack

Decision Version : Given n objects
with weights w_1 w_2 .. w_n
profits p_1 p_2 .. p_n
and Knapsack of size B

Is there a subset to achieve profit K

Knapsack $\in NP$

Subset Sum Problem

Std case : Given numbers
x_1 x_2 .... x_n

Can we partition them into two subsets
S_1 and S_2 s.t. $\sum_{x \in S_1} x = \sum_{y \in S_2} y$

$S_1 \cap S_2 = \phi$