hardness complexity

sum $O(n)$

hardest : incomputable (undecidable)

Halting problem

Given a program P and an input x, does P halt on x?

Assume ∃ program Halt :

$$\text{Halt}(P, x) = \begin{cases} \text{yes} & \text{if } P \text{ halt on } x \\ \text{no} & \text{otherwise} \end{cases}$$

Diagonal (P)

1. if Halt (P, P)
2. go to step 1

$$\text{Diagonal}(P) \begin{cases} \text{halt} & \text{if } P \text{ loops on } P \\ \text{loop} & \text{if } P \text{ holds on } P. \end{cases}$$

Diagonal halts on P if and only if P loops on P.

let P = diagonal ⇒ contradiction

Problem: $\begin{cases} \text{incomputable} \\ \text{computable} \end{cases}$ $\begin{cases} \text{complexity class} \\ \text{P. NP. EXP. PSPACE, co-NP. RP} \ldots \end{cases}$

1. Given a weighted graph G, and vertice s and t,
what is the shortest path from s to t

2. .. ..
what is the length of shortest .. ?

3. Given G, s, t, and integer K,
is there a path from s to t whose length $\le K$.
$\Rightarrow$ decision problem
yes or no

$1 \Rightarrow 2$ $\therefore$ $1 \geqslant 2$
$2 \Rightarrow 3$ $\therefore$ $2 \geqslant 3$
$\therefore$ $1 \geqslant 2 \geqslant 3$
$3 \Rightarrow 2$: bisection, loop.
$2 \Rightarrow 1$: delete a certain edge, if the answer of 2 changed, then this is in the shortest
$\therefore$ $3 \geqslant 2 \geqslant 1$

$\langle G, s, t, K \rangle \xrightarrow{\text{encoding}}$ binary string
define a set X = { encodings of $\langle G, s, t, K \rangle$ for whose answer is yes }

change to
$3 \Leftrightarrow$ Given a string s, is $s \in X$?

decision problem $\Longleftrightarrow$ language

instance $\langle G, s, t, K \rangle$ $\longleftrightarrow$ string $s = 0110 \ldots$

An algorithm A is a program, when given a string s, return yes or no $\rightarrow A(s)$

An algorithm A solves a problem X, if for any string s
$A(s) =$ yes if and only if $s \in X$

An algorithm A has a polynomial running time if there is a polynomial function $P(\ )$ so that for every string s, A terminates on s within $P(|s|)$ steps

P is the set of all problems for which there exist a polynomial time algorithm

$(\bar{X}_1 \vee X_2 \vee X_3) \wedge (X_1 \vee \bar{X}_2 \vee X_3) \wedge (X_1 \vee X_2 \vee X_4)$ to make the expression T
↳ satisfiability SAT

hint $(X_1 = 1,\ X_2 = 1,\ X_3 = 0,\ X_4 = 1)$

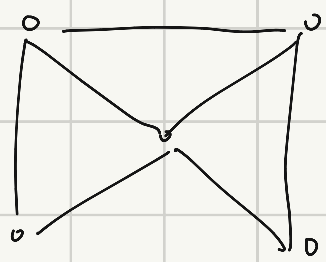
We say B is an efficient verifier for problem X if
1) B is a polynomial algorithm that takes two argument s and t
2) there exists a poly function $P(\ )$ so that for every string s, $s \in X$ if and only if $\exists$ a string t such that $B(s,t) =$ yes $\quad |t| \le P(|s|)$

B(s,t):

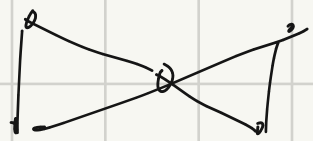
1. evaluate s under t.
2. return yes if s is satisfied by t, otherwise no

Hamiltonian Cycle Problem.

Given a $G = \langle V, E \rangle$, is there a simple cycle that visit all vertices exactly once



Yes



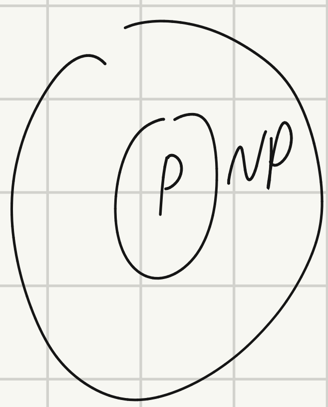
No

hint: the cycle

B: execute according to hint

NP: is the set of all problem for which there exists an efficient verifier

Lemma: $P \subseteq NP$



$X \in P$, $\exists A$ solves X

B(s,t)

1. run A on s
2. return the result

$P = NP$? unknown

reduction.

$X \leq_p Y$

① reduction X to Y in poly

② Y is harder than X

Problem: X Y

$f \rightarrow$ polynomial time

input $I_x \xrightarrow{\hspace{3em}} I_y = f(I_x)$

$I_x \in X$ iff $I_y = f(I_x) \in Y$

$\exists$ polynomial-time algorithm A_Y solves Y

$I_x \in X$?

$$I_x \rightarrow \boxed{f} \rightarrow f(I_x) \rightarrow \boxed{A_Y} \rightarrow A(f(I_x))$$

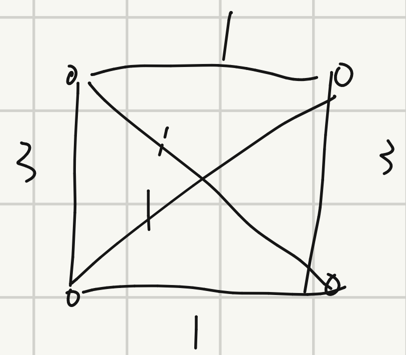
$\mathrm{poly}(|I_x|)$ $\mathrm{poly}(|f(I_x)|)$

$$\mathrm{poly}(|I_x|) + \mathrm{poly}(\mathrm{poly}(|I_x|)) = \mathrm{poly}(I_x)$$

If $X \leq_p Y$ and $Y \in P$, then $X \in P$

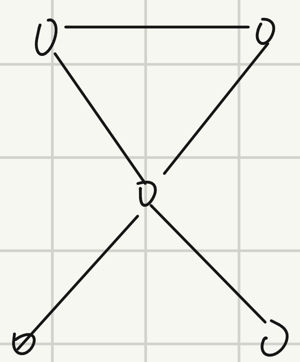
Traveling Salesman Problem (TSP)

Given a weighted complete graph $G = (V, E)$ and an integer K, is there a simple cycle that visits all vertices exactly once and with total cost $\leq K$



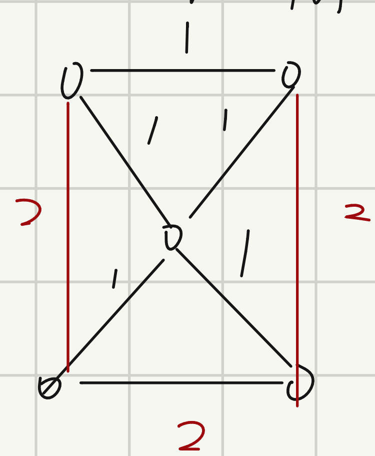
$HCP \leq_P TSP$

$G = (V, E)$



$G' = (V', E')$ and K'

$K' = |V|$



if G has a hamiltonian cycle,

G' has a solution with cost at most $|V|$

if G has no Hamiltonian cycle

every solution of G' has cost at least $|V|+1$

clique problem

Given a graph $G = (V, E)$, and an integer, does G has a complete subgraph with at least K nodes?