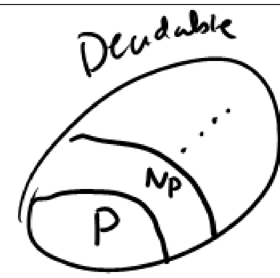
Lecture #25

P and NP

Let $M$ be a DTM that halts on all inputs.




The time-complexity of $M$ is $f: \mathbb{N} \to \mathbb{N}$

$$f(n) = \max_{\substack{\text{all word } w \\ \text{of length } n}} \left\{ \begin{array}{c} \text{time taken (\#steps)} \\ \text{by } M \text{ on } w \text{ to halt} \end{array} \right\}$$

If $f : \mathbb{N} \to \mathbb{R}^+$

$g : \mathbb{N} \to \mathbb{R}$

then $f = O(g)$

if $\exists c, n_0 \in \mathbb{N}$

s.t $\forall n \geq n_0 \quad f(n) \leq c \cdot g(n)$.

---

$f_1(n) = 3n^2 + 2n + 9 \quad ; \quad g_1(n) = n^2$

$g_2(n) = n^3$

$f_1 = O(g_1)$

$f_1 = O(g_2)$

$$n \log_{10} 5n + n \log_{10} \log_{10} n$$
$$= O(n \log_2 n)$$

---

$t : \mathbb{N} \to \mathbb{R}^+$

$\mathrm{TIME}(t(n))$ : all languages $L$ that are decidable by some DTM working in time $O(t(n))$

$\mathrm{NTIME}(t(n))$ : all languages $L$ that are decidable by some NTM working in time $O(t(n))$

max time taken over any non-det path

$\mathrm{TIME}(n^2)$ : problems that can be solved det by a TM in quadratic time.

$$L = \{ 0^n 1^n \mid n \geq \cancel{\text{}} 0 \}$$

1. DTM

- Do ~~#~~ passes on the tape, removing one 0 and one 1 in each pass
- If 0s remain & 1's don't or 1's remain & 0's don't reject

else accept.

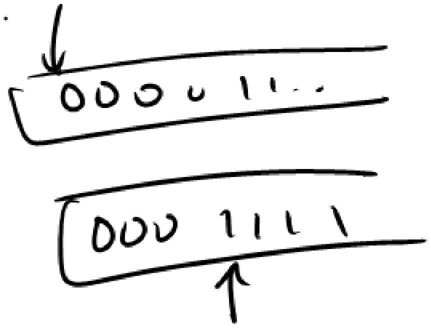
$O(n^2)$ algorithm.

$O(n \log n)$ time is possible.

2. DTM with 2 tapes

Linear-time algm.

Go right on both tapes crossing off a 0 with a 1.




PTIME – TMs are fairly robust

$$P = \bigcup_{k \geq 0} \mathrm{TIME}(n^k).$$

---

$n^3$ $\quad$ $n = 1000$ $\quad$ $n^3 = 1$ billion

$2^n$ $\quad$ $2^n >$ # atoms in universe.

$n^{100}$

$O(n^2)$ algm is much slower $O(n)$ algm.

Every multitape TM with running time $t(n)$
is equivalent to a singletape TM
with running time $O(t^2(n))$

This is also true for RAM models.

$$P : \bigcup_{k \geq 0} TIME(n^k)$$

~~NTIME~~

$$NP : \bigcup_{k \geq 0} NTIME(n^k)$$

$$P \subseteq NP \longrightarrow P \subsetneq NP$$



$$P \subseteq NP \longrightarrow P = NP$$

# NP using certificates


A **verifier** for a language $L$ is an algorithm $V$

$$L = \{ w \mid V \text{ accepts } \langle w, c \rangle \text{ for some string } c \}$$

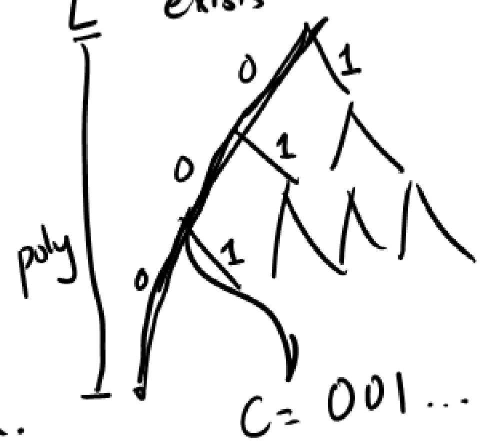
If $w \in L$ then $\exists c.\ \langle w, c \rangle \in L(V)$

If $w \notin L$ then $\forall c.\ \langle w, c \rangle \notin L(V)$.

A polytime verifier is a verifier that runs in polytime (and hence $|c| \le poly(|w|)$)

A language $L$ is in NP iff $L$ has a polytime verifier

**Proof.** ($\Rightarrow$) NTM for $L$ exists



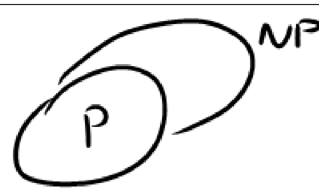

A verifier for $L$

- Input $\langle w, c \rangle$
- Simulates NTM on $w$ using $c$ to resolve non-determinism.

($\Leftarrow$) $L$ has a verifier

NTM for $L$:

- Guess certificate $c$
- Simulate verifier on $\langle w, c \rangle$




**P**: 1) Reachability in a graph $\langle G, s, t \rangle$

$s \rightsquigarrow t$

$O(n)$ algorithm

2) CFG membership.

3) Euclid's algm for relative prime.

4) Primes

5) LP

$$A\bar{x} \leq B \qquad \max f(\bar{x})$$

$$5x+3y \leq 18$$
$$9y \leq 7$$
$$5x+7z \geq 15$$
$$\max 3x+5y$$

NP

NPC • Boolean satisfiability (SAT, Cook-Levin)

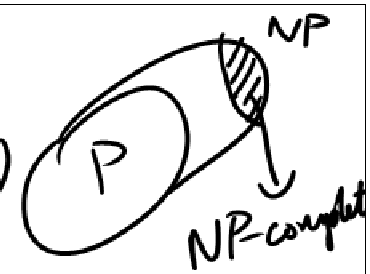
NPC • Hamiltonian path in a graph

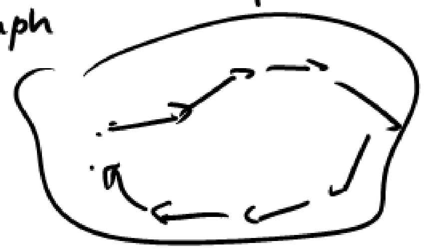
not known to be NPC • Graph Isomorphism

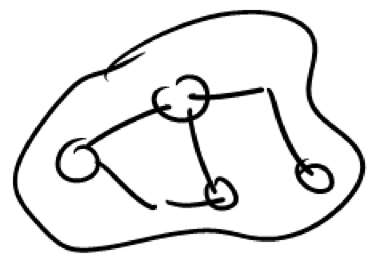
NPC • K-clique

NPC • TSP

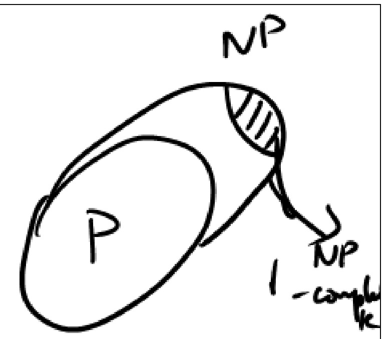
• Vertex cover





## Polytime reducibility.




$f: \Sigma^* \to \Sigma^*$ is
polynomial-time computable
if there is a DTM that computes $f(w)$,
given $w$, in time time $n^k$
where $n = |w|$, $k$ is a constant.

$A$ is polynomial-time mapping reducible

($A$ is polynomial-time reducible) to $B$

denoted $A \leq_p B$

if there is a polynomial time computable $f: \Sigma^* \to \Sigma^*$

such that $\forall w \in \Sigma^*$

$w \in A$ iff $f(w) \in B$.

If $B \in P$ and $A \leq_p B$ then $A \in P$

If $B \in NP$ and $A \leq_p B$ then $A \in NP$

A language $L$ is NP-complete if

1) $L \in NP$
2) For every $A \in NP$, $A \leq_P L$

---

So if $L$ is NP-complete and $L \in P$

then $NP \subseteq P$

i.e. $P = NP$.

If $P = NP$, $L \in P$
If $P \neq NP$, $L \notin P$ $\Big\}$ $L \in P$ iff $P = NP$.