L22 2022-11-09

Polynomial Time

def $P = \bigcup_c \mathrm{DTIME}(n^c)$ $\quad NP = \bigcup_c \mathrm{NTIME}(n^c)$

$L \in P$ if $\exists c$ and M deciding L in time $O(n^c)$.

$L \in NP$ " N " " .

To understand NP, we'll give an alternative def.

Recall the TM for ¬PALINDROME:

N: guess i
  if x[i] ≠ x[n-i-1]: accept
  reject.

We can split it in two parts:

	Nondeterminism	Computation
1. guess	✓	✗
2. verify	✗	✓

In fact, any NTM can be split in the same way, by moving all guesses to the beginning.

Obs we can only make a polynomially long guess.

Hence we define:

$L \in NP$ if $\exists$ TM V running in poly-time st

$x \in L$ iff $\exists^P y$ st $V(x,y)$ accepts.

short for $\exists y, |y| = \mathrm{poly}(x)$

y is the witness aka certificate.
V is the verifier.

obs V must always reject if $x \notin L$.

Hence we can think as P = "efficiently computable" and NP = "efficiently verifiable".

Examples:

P	NP
graph completeness	clique
shortest path	longest path
2-SAT	3-SAT
ckt eval	ckt-SAT
linear equations	quadratic equations
primality	factoring