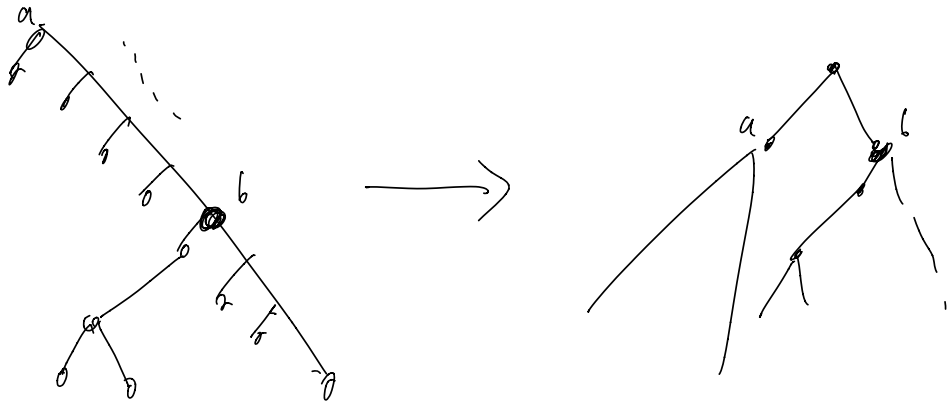


P, L, E, EXP

DTIME(t(n))

1. How does randomness help?
2. How does non-determinism help?
3. How does quantum help?
4. Relationship between complexity of specific functions?

NP

Compute f if

$f(x) = 1 \Rightarrow$ some guesses leading to output 1

$f(x) = 0 \Rightarrow$ no guesses lead to output 1.

NTIME(t(n))

Thm: If r, t are time constructible $r(n+1) = o(t(n))$ then

$NTIME(r(n)) \neq NTIME(t(n))$.

Subset sum

Input: A list of $a_1, \ldots, a_k, t \in \mathbb{N}$

Output: 1 iff $\exists S \sum_{i \in S} a_i = t$

Composite

Input: $x \in \mathbb{N}$

Output: 1 iff x is composite

Independent set (set of vertices not including any edges)

Input: graph G, $k \in \mathbb{N}$

Output: 1 iff G has an id set of size $\geq k$.

Def: $f: \{0,1\}^* \to \{0,1\}$ is in NP if $\exists$ poly p and a polytime machine V s.t. $\forall x \in \{0,1\}^*$

$f(x) = 1 \Leftrightarrow \exists w \in \{0,1\}^{p(|x|)}$

$V(x,w) = 1$.

FACT $P \subseteq NP \subseteq EXP$

OPEN: Is $P = NP$?

Def: f is polytime reducible to g if $\exists$ poly time computable h s.t
$f(x) = g(h(x))$

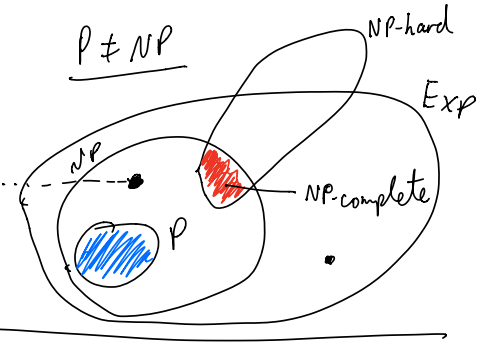
$f \leq_p g$.

Def: f is NP-hard if $g \leq_p f$ for every $g \in NP$. We say f is NP-complete if f is NP-hard and $f \in NP$.

P = NP

P = NP = NP-complete

P ≠ NP



$\exists$

Circuit-SAT:

Input: circuit C

Output: 1 iff $\exists y$ s.t $C(y)=1$

Circuit-SAT $\in$ NP ✓

Circuit-SAT is NP-hard

If $g \in NP$, let $V(x,w)$ be the verifier for g.

$h(x)$

compute circuit $C(w)$ s.t $V(x,w) = C(w)$.

output C.

$g(x) = $ Circuit-SAT $(h(x))$.

$f \leq_p g \leq_p h \Rightarrow f \leq_p h$

Circuit-Sat $\leq_p$ 3-SAT

3-SAT formula

$(x_1 \vee \bar{x_2} \vee x_3) \wedge (x_4 \vee \bar{x_1} \vee x_2) \wedge \cdots$

clause

3-SAT Input: ϕ

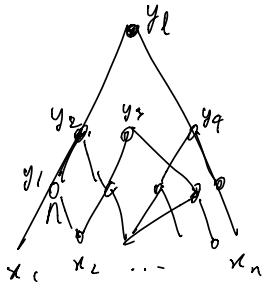
Output: 1 iff $\exists y$ s.t $\phi(y) = 1$.

FACT: $\forall f: \{0,1\}^n \to \{0,1\}$ $\exists$ n-SAT formula of size $O(n 2^n)$ computing f.

x_1	x_2	$f(x)$
0	0	0
0	1	1
1	0	0
1	1	1

$(x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2)$

Circuit-SAT $\leq_p$ 3-SAT



$\xrightarrow{h}$ $(\cdots) \wedge (\cdots) \wedge (\cdots) \cdots$

ensure y_1 is correct

$y_1 = x_1 \wedge x_2$

formula is satisfiable iff Circuit is sat.

∃ polytime algo. for 2-SAT

$(a \vee \bar{b})$

$b \Rightarrow a$

t $\quad a_1 \cdots a_\ell$

2^t $\quad 2^{a_1} \cdots 2^{a_2} \cdots 2^{a_\ell}$

X

Primes in P

x