The Class NP

- Do T.S.P., Sub-Set Sum*, MAX-CUT, 3SAT, ...
  have a polytime algorithm?

- Not known.

- They do have a <u>non-deterministic</u> polytime algorithm.

<u>Def</u> A non-det TM M is a tuple

$$M = (Q, \Sigma, \Gamma, \delta, q_{start}, q_{accept}, q_{reject})$$

where $\delta : Q \times \Gamma \longrightarrow (Q \times \Gamma \times \{L, R\})^2$.

<u>I.e.</u> An instruction is of form

$$(q, a) \begin{array}{l} \nearrow (q', a', L) \\ \text{or} \\ \searrow (q'', a'', R). \end{array}$$

Two (or more or none) choices at each step.

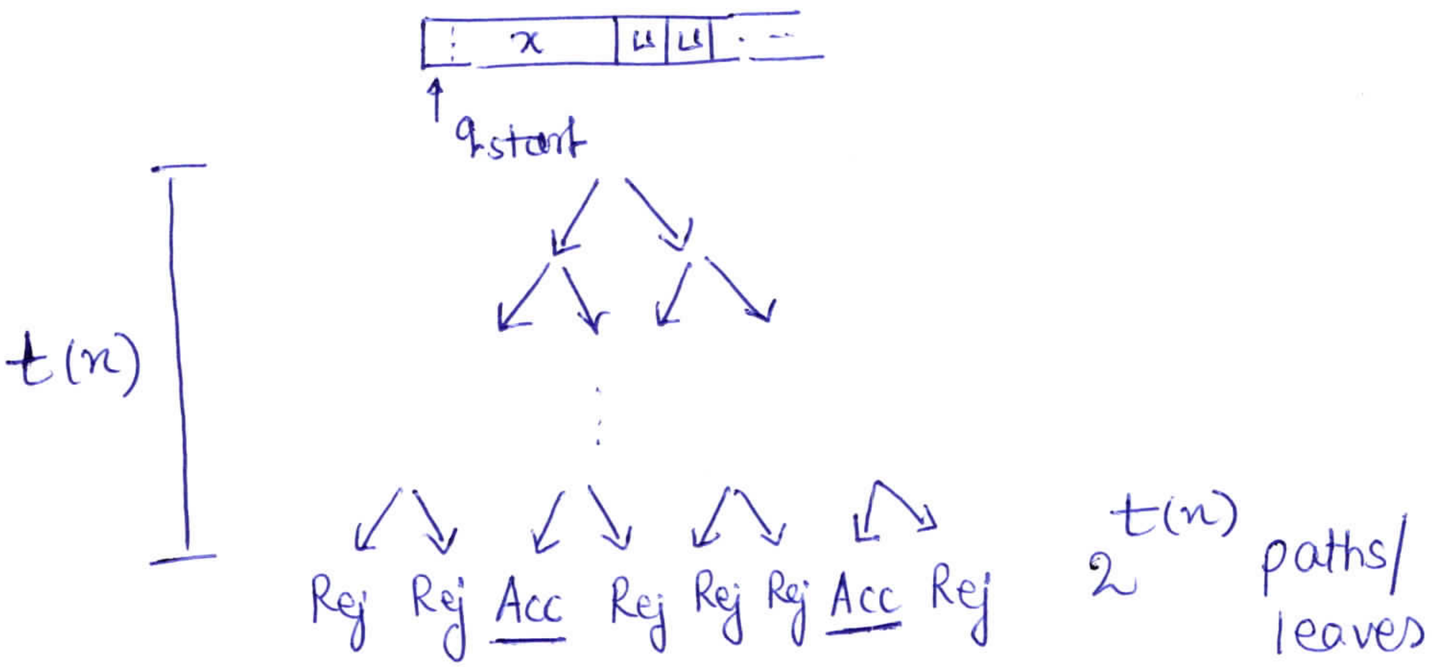
<u>Note</u>. We'll assume a time bound $t(n)$.

Conventions (w.l.o.g.)

- Machine halts in time/#steps exactly $t(n)$.

  Running time
- Exactly two choices at each step.

Computation of a Non-det TM



Def Input $x \in \Sigma^*$ is accepted by a NTM M if on input x, M has at least one computation that accepts.

Def A language L is accepted by a NTM M in time $t(n)$ if

- $\forall x \in \Sigma^*, |x| = n$, M runs for time (at most) $t(n)$.

- $x \in L \Rightarrow$ M has at least one computation on x that accepts.
- $x \notin L \Rightarrow$ Every computation of M on x rejects.

Def $NTIME(t(n)) :=$ Class of languages accepted by NTMs in time $O(t(n))$.

Note $NTIME(t(n)) \subseteq DTIME(2^{t(n)})$.

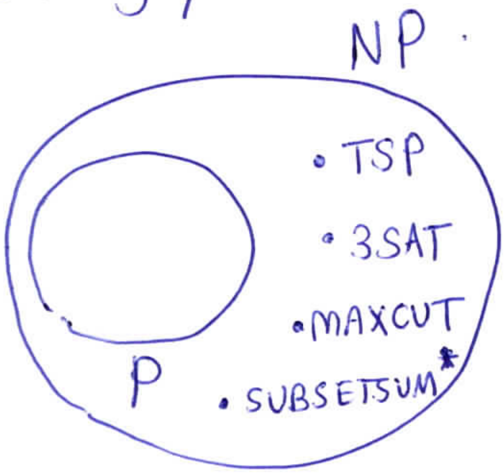
Proof In det time $2^{t(n)}$, all $2^{t(n)}$ paths of a NTM can be simulated and then one checks if at least one path accepts. ∎

Def

$$NP = \bigcup_{k=1,2,3,\ldots} NTIME(n^k).$$

Clearly
- $DTIME(t(n)) \subseteq NTIME(t(n))$.
- $P \subseteq NP$.

Seemingly



- P vs NP.
- Widely believed that $P \subsetneq NP$.

Problems in NP

All problems in P

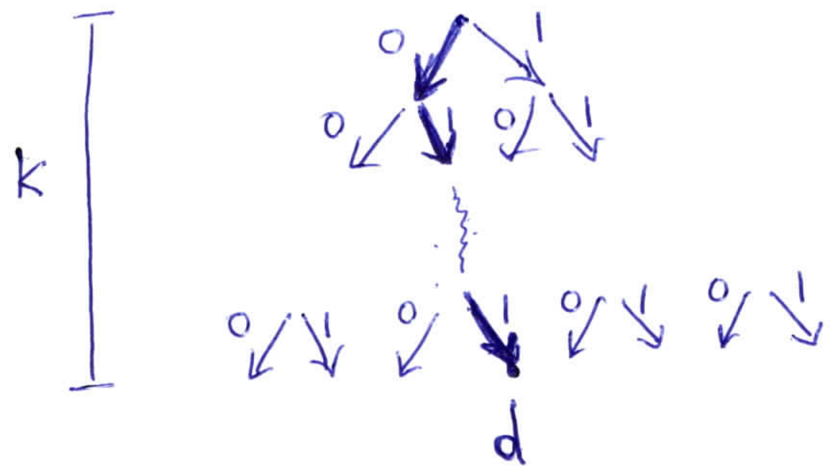
COMPOSITE $\in$ NP

$$\text{COMPOSITE} = \left\{ \langle n \rangle \;\middle|\; \begin{array}{l} \langle n \rangle \text{ is binary rep of } n, \\ n \text{ is composite} \end{array} \right\}$$

Note that if $k = \#$ bits in n then the input size is k.

Following polytime NTM accepts COMPOSITE.

M := " On input $\langle n \rangle$,
let $k = \#$ bits in n.
Non-deterministically choose a k-bit integer d.
Accept if $2 \leq d < n$ and d divides n.
Reject otherwise. "



Note that one can check in (det) time poly(k) whether $2 \le d < n$ and whether d divides n. ▨.

$\langle n \rangle \in \text{COMPOSITE} \Rightarrow \exists\, d,\ 2 \le d < n,\ d \mid n$

$\Rightarrow \exists$ at least one accepting computation.

$\langle n \rangle \notin \text{COMPOSITE} \Rightarrow \nexists\, d,\ 2 \le d < n,\ d \mid n$

$\Rightarrow$ Every computation rejects.

▨

d = "witness". that $\langle n \rangle \in$ COMPOSITE.

= "proof"

▨

SUBSET-SUM $\in$ NP

$$\text{SUBSET-SUM} = \left\{ \begin{array}{l} (a_1, a_2, \ldots, a_n ; t) \mid \\ a_1, a_2, \ldots, a_n, t \text{ are } n\text{-bit integers}, \\ \exists S \subseteq \{a_1, \ldots, a_n\} \text{ s.t. } \sum_{i \in S} a_i = t \end{array} \right\}.$$

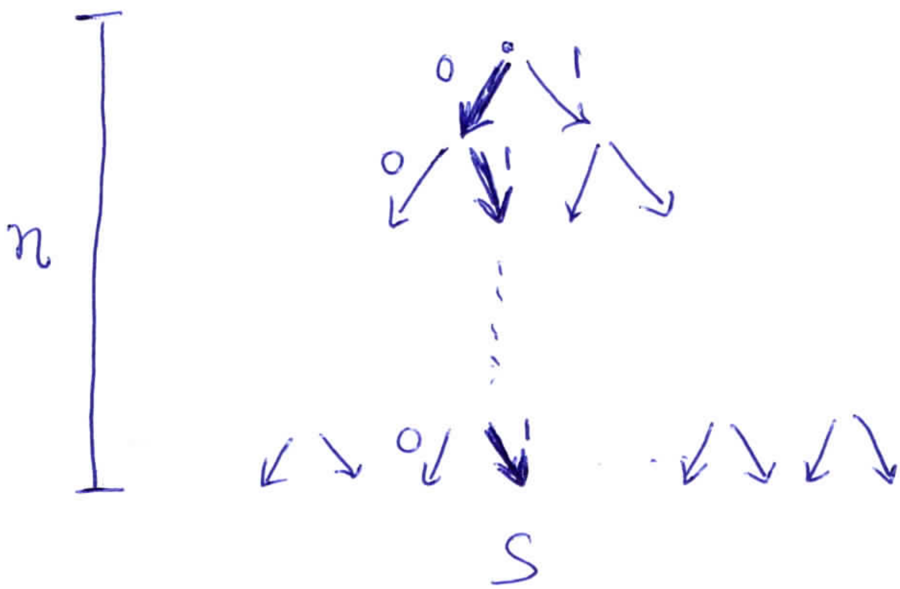
Following poly(n)-time NTM M accepts SUBSET-SUM.

$M :=$ " On input $(a_1, \ldots, a_n ; t)$,

Non-det choose a subset $S \subseteq \{a_1, \ldots, a_n\}$.

Accept if $\sum_{i \in S} a_i = t$.

Reject otherwise "



$S = \{ i \mid 1 \le i \le n, \text{ Choice '1' is made at step } i \}$.

Traveling Salesperson $\in$ NP

$TSP = \{ \langle G, l \rangle \mid$ G is a weighted graph and has a tour of length $\leq l \}$

Following polytime NTM M accepts TSP.

M := "On input $\langle G, l \rangle$,

Non-det choose a tour σ.

If weight of σ is $\leq l$, accept.

Reject otherwise"

PRIMES $\in$ NP

$PRIMES = \{ \langle n \rangle \mid$ n is a k-bit integer that is a prime. $\}$.

- How can I prove to you that n is a prime (in time poly(k))?
- Hmm.... Highly non-trivial!

As it turns out PRIMES $\in$ P!

Def A (polytime) NTM M is said to be "guess & verify" machine if there is a polytime **det** machine $M_{verifier}$ and

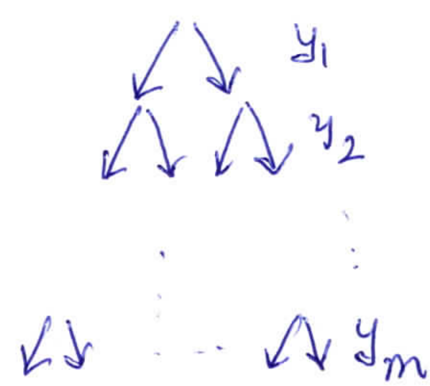
M := " On input x, $|x| = n$,

"Guess" { Non-det select a string y, $|y| = n^{O(1)}$,

"verify" { Run $M_{verifier}$ on pair (x, y). Accept iff $M_{verifier}$ accepts "

Note - All NTMs we saw are guess & verify m/c's.

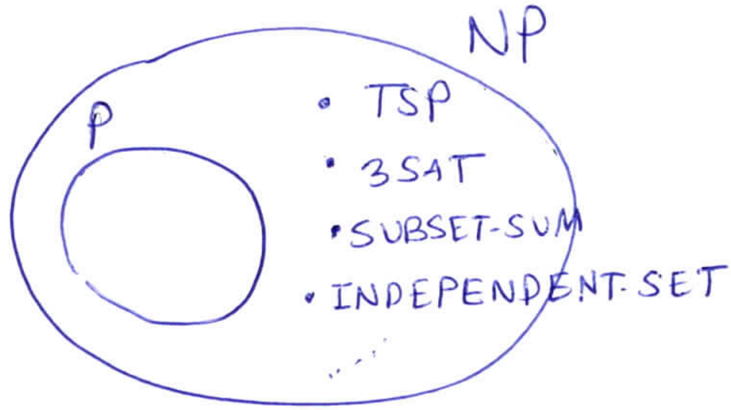
- W.l.o.g. one can assume that NTMs are guess & verify m/c's (by first non-deterministically selecting string y denoting all choices and then simulating the path along those choices.



- m = run time
- $y_1 y_2 \cdots y_m \in \{0,1\}^m$.

$\boxed{y = \text{"witness"/"proof"}}$

Reductions and NP-completeness



- We don't know if TSP, 3SAT, SUBSET-SUM, --- have a (det) polytime algorithm.
- But we know that if one of them does, so do all others!
- So these are the "hardest problems" in NP!

Reductions

<u>Def</u> A (det) TM M with output has an extra, one-way output tape to write on. On input x, the contents of the output tape after the m/c halts is its output $M(x)$.

<u>Note</u> If M is polytime then $|M(x)| = |x|^{O(1)}$.

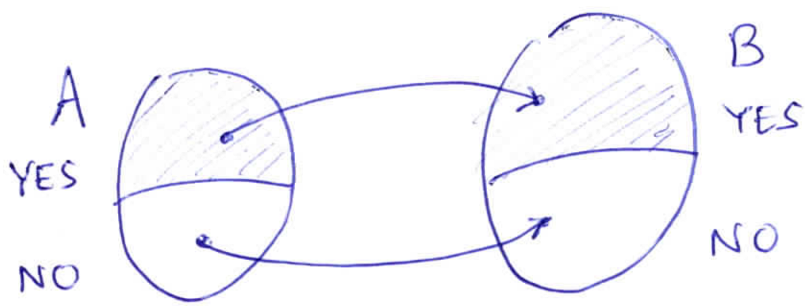
Def A language A reduces in polynomial time to a language B if there is a polytime det TM M with output s.t. $\forall x \in \Sigma^*$, on input x, M outputs $y = M(x)$

and $x \in A \iff y \in B$.

This is denoted as $A \le_p B$.

Claim If $A \le_p B$ and $B \in P$ then $A \in P$.

Proof



M = Reduction

$x \rightsquigarrow y$

Suppose M_B decides B in polytime.

Time n^c

M_{red} is the reduction A to B.

Time n^k

Then M_A decides A in time n^{CK} as:

M_A := " On input x, $|x| = n$,

Run M_{red} on x to output y.

Using M_B, decide if $y \in B$.

Accept iff M_B accepts y. "

<u>Correctness</u> $x \in A \iff y \in B$.

<u>Runtime</u> - $|y| \leq |x|^K = n^K$.

- Runtime of M_B on y is

$|y|^C \leq (n^K)^C = n^{C \cdot K}$. $\blacksquare$

<u>Note similarly</u> If $A \leq_p B$, $B \leq_p C$

Then $A \leq_p C$.

<u>Proof</u> $A \xrightarrow{red} B \xrightarrow{red} C$

$x \rightsquigarrow y \rightsquigarrow z$

$x \in A \iff y \in B \iff z \in C$. $\blacksquare$