Def 7.12: The class $P$ consists of languages that are decidable in polynomial time, i.e.,

$$P = \bigcup_k \text{TIME}(n^k)$$

Example:

$\text{PATH} = \{ <G,s,t> \mid G \text{ is a digraph that has a path from } s \text{ to } t \}$

- BFS, DFS, Dijkstra
  - Takes $O(1|V|+|E|)$ steps on a random access machine, where $|V|$ - #vertices, $|E|$ - #edges
  - $n = \text{the TM length of input}$ is the length of encoding of $<G,s,t>$, about $O(|E|\cdot \log(|V|) + |V|)$
  - Then the TM running time: about $O(n^2)$ because not random access
Def 7.12: The class P consists of languages that are decidable in polynomial time, i.e.,

\[ P = \bigcup_k \text{TIme}(n^k) \]

Example:

\[ \text{RELPRIME} = \{ <x,y> \mid x \text{ and } y \text{ are relatively primes} \} \]

Euclid algorithm:

- Length of the input: \( n = O(\log x + \log y) \)
- Euclid’s algo run. time: polynomial in \( \log x + \log y \) (because the number of iterations is logarithmic)
Def 7.12: The class $P$ consists of languages that are decidable in polynomial time, i.e.,

$$P = \bigcup_k \text{TIME}(n^k)$$

Example:

$\text{PRIME} = \{ <x> \mid x \text{ is a prime} \}$

for $i$ from 2 to $\sqrt{x}$
    check if $i$ divides $x$ - if yes, return NO PRIME (reject)
return PRIME (accept)

running time $O(\sqrt{x})$ (random access machine, no bit-counting per number)

length of input: $n = O(\log x) \rightarrow O(2^{n/2})$
The Class NP

Example:

\[ \text{HAMPATH} = \{ <G,s,t> \mid G \text{ is digraphs with Hamiltonian path from s to t} \} \]
The Class NP

Example:

\[
\text{COMPOSITES} = \{ x \mid x = pq, \text{ for some } p, q > 1 \}
\]
The Class NP

Def 7.18: A **verifier** for a language $A$ is an algorithm $A$, where

$$A = \{ w \mid V \text{ accepts } <w,c> \text{ for some string } c \}$$

Polynomial-time **verifier** runs in (deterministic) time polynomial in the length of $w$. The string $c$ is called the **certificate**, or **proof**, of the membership in $A$. 
The Class NP

Def 7.18: A **verifier** for a language $A$ is an algorithm $A$, where

$$A = \{ w \mid V \text{ accepts } <w,c> \text{ for some string } c \}$$

**Polynomial-time verifier** runs in (deterministic) time polynomial in the length of $w$. The string $c$ is called the **certificate**, or **proof**, of the membership in $A$.

Def 7.19: **NP** is the class of languages that have polynomial time verifiers.
Thm 7.20: A language is in NP iff it is decided by some nondeterministic polynomial-time TM.

Def 7.21: NTIME(t(n)) = { L | L is a language decided by a O(t(n))-time nondeterministic TM }.

Thus, \[ \text{NP} = \bigcup_k \text{NTIME}(n^k) \]
The Class NP

Thm 7.20: A language is in NP iff it is decided by some nondeterministic polynomial-time TM.

Def 7.21: $\text{NTIME}(t(n)) = \{ L | L \text{ is a language decided by a } O(t(n))\text{-time nondeterministic TM} \}$.

Thus, $\text{NP} = \bigcup_k \text{NTIME}(n^k)$
The Class NP

Example:

\[
\text{CLIQUE} = \{ <G,k> \mid G \text{ is undirected graph with a } k\text{-clique} \}
\]

\[\begin{array}{ccc}
G & & \\
\text{k = 6} & \text{NO} & \\
\text{k = 4} & \text{YES} & \\
\end{array}\]

\[\text{CLIQUE} \in \text{NP}\]

- \(k\) vertices
- (will guess the clique)
- check all possible \(k\)-cliques
- then verify if all edges are present

\(\checkmark\)
The Class NP

Example:

\[ \text{SAT} = \{ \langle \phi \rangle \mid \phi \text{ is a satisfiable Boolean formula} \} \]

\[ \phi = x_1 \lor x_2 \land (x_3 \lor x_4 \lor x_2) \land \overline{x_5} \]

\[ \begin{array}{ccc}
T & T & T \\
& & x_5 = F \\
\end{array} \]

- does there exist True/False assignment to \( x_i \)'s s.t. \( \phi = \text{true} \)?

\[ \text{SAT} \in \text{NP} \]

\[ c = T/F \text{ assignment to the } x_i \text{'s} \]

\[ V = \text{verifies if } \phi = \text{true} \]
The Class NP

Wrapping up:
- P - exists polynomial-time algorithm
- NP - exists polynomial-time verifier

BIG open problem:

Is P = NP ???

Note: also exists a class coNP, the class of complements of problems in NP (e.g. CLIQUE$^c$, “is every clique of a given graph of different size than k?”). We do not know if NP = coNP.