TIME COMPLEXITY

Let \( t : \mathbb{N} \rightarrow \mathbb{N} \) be a function. Time complexity class
\[
\text{TIME}(t(n)) = \{ L \mid L \text{ is a language decided by an } O(t(n)) \text{ TM} \}
\]
\[A = \{ 0^k 1^k \mid k \geq 0 \} \]
We know TM such that \( A \in \text{TIME}(n^2) \)

\( M_2: \)
- Repeat as long as some 0's & some 1's remain on tape
  - Scan the tape. If total # 0's & 1's remaining is odd, \( \text{REJECT} \)
  - Scan the tape, crossing off every other 0 starting with the first, & then crossing off every other 1 starting with the first.
- If no 0's & no 1's remain on tape, \( \text{ACCEPT} \) else \( \text{REJECT} \).

With \( M_2 \), \( A \in \text{TIME}(n \log n) \).
With a 2-tape TM \( M_3 \), \( A \in \text{TIME}(n) \).

Computability: all reasonable models of computation are \( \leftrightarrow \)
Complexity: choice of model affects time complexity.

\( \text{Th:} \) Each \( t(n) \) time multitape TM has an \( \leftrightarrow O(t^2(n)) \) time single tape TM.

\( \text{Th:} \) Each \( t(n) \) time non-deterministic single tape TM has an \( \leftrightarrow 2^{O(t(n))} \) time deterministic single tape TM.
**Def.** $P$ is the class of languages that are decidable in polynomial time on a deterministic 1-tape TM.

$$P = \bigcup_{k \geq 1} \text{TIME}(n^k)$$

1) $P$ is invariant for all models of computation that are polynomially equivalent to a 1-tape TM.

2) $\leftrightarrow$ class of problems realistically solvable on a computer.

$\text{PATH} = \{ \langle G, s, t \rangle \mid \text{directed path in } G \text{ from } s \text{ to } t \}$

$\text{PATH} \in P$

$M(\langle G, s, t \rangle)$:

1) Mark $s$

2) Repeat until no additional nodes are marked

3) Scan edges of $G$. If edge $(a, b)$, mark $b$

4) If $t$ is marked, then ACCEPT, else REJECT.

This $M$ uses a breadth-first search. Running time?

$\text{RELPRIME} = \{ \langle x, y \rangle \mid x \text{ and } y \text{ are relatively prime} \} \in P$

**Euclid's GCD** $\langle x, y \rangle$:

1) Repeat until $y = 0$

2) $x \leftarrow x \mod y$

3) Exchange $x$ & $y$

4) Output $x$.

Running time: $\min(\log_2 x, \log_2 y) * x$?
THE CLASS NP

Hamiltonian path = directed path that goes thru each node once

\[ \text{HAMPATH} = \{ \langle G, s, t \rangle \mid \text{Hamiltonian path from } s \text{ to } t \text{ in } G \} \]
- polynomial time solution unknown
- has polynomial verifiability

HAMPATH not polynomially verifiable

\[ \text{COMPOSITES} = \{ x \mid x = pq, \text{ for integers } p, q > 1 \} \]

Defn: A verifier for a language \( A \) is an algorithm \( V \), where
\[ A = \{ \omega \mid V \text{ accepts } \langle \omega, c \rangle \text{ for some (certificate) string } c \} \]
A poly-time verifier runs in poly-time in length of \( w \)
\( A \) is poly-verifiable if it has a poly-time verifier.
Verifiers use a certificate (proof) \( c \) to verify \( w \in A \).

\[ \text{For } w = \langle G, s, t \rangle \in \text{HAMPATH}, c = \text{the Hamiltonian path} \]
\[ \text{For COMPOSITES, } c = \text{a divisor of } x \]

Defn: \( \text{NP} \) = class of languages that have poly-time verifiers.
(Non-deterministic polynomial)

Theorem: A language is in \( \text{NP} \) if it is decided by a
non-deterministic poly-time TM.
\[ \text{N} \text{TM simulates verifier by guessing certificate} \]

\[ \langle w, c \rangle \rightarrow V \rightarrow A \]

\[ \langle w \rangle \rightarrow \text{guess} \ c \quad \text{length}\ n^k \]

\[ \text{Verifier simulates the NTM by using the accepting branch as the certificate} \]

\[ \langle w, c \rangle \rightarrow \langle w \rangle \rightarrow \langle c \rangle \rightarrow N \rightarrow V \rightarrow A \rightarrow A \]

\[ \text{treat each symbol of } c \text{ as a description of the non-deterministic choice at each step} \]

**Defn:** \( \text{NTIME}(t(n)) = \{ L \mid L \text{ decided by a } O(t(n)) \text{ time NDTM} \} \).

**Cor:** \( \text{NP} = \bigcup_{k} \text{NTIME}(n^k) \)

**Defn:** Clique in an undirected graph = subgraph where each pair of nodes has an edge connecting them.
**Th:** CLIQUE ∈ NP

\[ V(\langle G, k \rangle, c) \]

**SUBSET-SUM = \{ \langle S, t \rangle \mid S = \{ x_1, x_2, \ldots, x_k \} \text{ and for some } \{ y_1, y_2, \ldots, y_e \} \subseteq S, \text{ we have } \sum y_i = t \} \]

// multi-sets allowed

**Th:** SUBSET-SUM ∈ NP

\[ V(\langle S, t \rangle, c) \]

**CLIQUE, SUBSET-SUM:** not obviously members of NP

\[ \text{NP} = \text{co-NP} ? \]

**P:** membership can be decided polynomially

**NP:** "verified"

\[ P \subseteq \text{NP} ; \ P = \text{NP} ? \]

\[ \text{NP} \subseteq \text{EXPTIME} = \bigcup_k \text{TIME}(2^{k^k}) \]

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

**3SAT = \{ \langle \phi \rangle \mid \text{"3cnf"} \}**

**Defn:** Language A is poly time reducible to lang B, \( A \leq_p B \), if a poly time computable func \( f: \Sigma^* \rightarrow \Sigma^* \) exists, where for each \( w \), \( w \in A \iff f(w) \in B \)

**Th:** if \( A \leq_p B \) and \( B \in \text{P} \), then \( A \in \text{P} \).
NP-Completeness

Defn: Lang B is NP-Complete if $B \in \text{NP}$ and

1) $\forall A \in \text{NP}, A \leq_p B$

Th: If $B$ is NP-complete & $B \in \text{P}$, then $P = \text{NP}$

SAT, 3-SAT $\in$ NP-Complete

Th: If $B \in \text{NP}$-Complete and $B \leq_p C$ for $C$ in $\text{NP}$, then $C \in \text{NP}$-Complete

E.g. CLIQUE $\in$ NP-Complete

Clique $\in \text{NP}$. Show 3-SAT $\leq_p$ CLIQUE

(k clauses) (k-)

Eq: $\phi = (x_1 \lor x_1 \lor x_2) \land (\neg x_1 \lor x_2 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_2)$

Show $\phi$ is satisfiable $\iff G$ has a k-clique