Plan for today

1. Recap most of the topics that we discussed in the course
2. Talk about any questions that you still have
P, NP, NP-completeness

**Definition (The classes P and NP)**

\[ P = \bigcup_{c \geq 1} \text{DTIME}(n^c) \quad \text{NP} = \bigcup_{c \geq 1} \text{NTIME}(n^c) \]

**Definition**

A *polynomial-time reduction* from \( L_1 \subseteq \{0, 1\}^* \) to \( L_2 \subseteq \{0, 1\}^* \) is a polynomial-time computable function \( f : \{0, 1\}^* \to \{0, 1\}^* \) such that for each \( x \in \{0, 1\}^* \) it holds that \( x \in L_1 \) if and only if \( f(x) \in L_2 \).

**Definition**

A problem \( L \subseteq \{0, 1\}^* \) is *NP-complete* if \( L \in \text{NP} \) and every problem \( L' \in \text{NP} \) can be polynomial-time reduced to \( L \).

**Theorem (Cook-Levin)**

*3SAT is NP-complete.*
### Theorem (Deterministic Time Hierarchy)

If $f, g$ are time-constructible functions such that $f(n) \log f(n)$ is $o(g(n))$, then:

$$\text{DTIME}(f(n)) \subsetneq \text{DTIME}(g(n))$$

### Theorem (Nondeterministic Time Hierarchy)

If $f, g$ are time-constructible functions such that $f(n + 1)$ is $o(g(n))$, then:

$$\text{NTIME}(f(n)) \subsetneq \text{NTIME}(g(n))$$

- So $P \subsetneq \text{EXP}$ and $NP \subsetneq \text{NEXP}$.

### Theorem (Baker-Gill-Solovay)

There exist oracles $A, B \subseteq \{0, 1\}^*$ such that $P^A = NP^A$ and $P^B \neq NP^B$. 
Space complexity

Definition (The classes L, NL, PSPACE, NPSPACE)

\[
L = \text{SPACE}(\log n) \quad \text{NL} = \text{NSPACE}(\log n) \\
PSPACE = \bigcup_{c \geq 1} \text{SPACE}(n^c) \quad \text{NPSPACE} = \bigcup_{c \geq 1} \text{NSPACE}(n^c)
\]

- NL-completeness: based on logspace reductions.

Theorem (Space Hierarchy)

If \( f, g \) are space-constructible functions such that \( f(n) = o(g(n)) \), then:

\[
\text{SPACE}(f(n)) \subsetneq \text{SPACE}(g(n))
\]

- So \( L \subsetneq \text{PSPACE} \).

Theorem (Savitch)

\( \text{PSPACE} = \text{NPSPACE} \).
Polynomial Hierarchy (PH)

**Definition (The classes \(\Sigma^p_i\))**

\(L \subseteq \{0, 1\}^*\) is in \(\Sigma^p_2\) if there exists a polynomial-time TM \(M\) and a polynomial \(q\) such that for all \(x \in \{0, 1\}^*\):

\[x \in L \iff \text{there exists } u_1 \in \{0, 1\}^{q(|x|)} \text{ such that for all } u_2 \in \{0, 1\}^{q(|x|)} \text{ it holds that } M(x, u_1, u_2) = 1.\]

(Similarly for all \(i \geq 1\).)

**Definition (The class PH)**

\[\text{PH} = \bigcup_{i \geq 1} \Sigma^p_i\]
Circuits and advice

**Definition (The class P/poly)**

\[ \text{P/poly} = \bigcup_{c \geq 1} \text{SIZE}(n^c) \]

P/poly consists of all \( L \subseteq \{0, 1\}^* \) that can be decided in polynomial time with polynomial-size advice \( \{\alpha_n\}_{n \in \mathbb{N}} \).

**Theorem (Karp-Lipton)**

*If \( \text{NP} \subseteq \text{P/poly} \), then \( \text{PH} = \Sigma^p_2 \).*
Definition (The classes BPP, RP, ZPP)

\[ BPP = \bigcup_{c \geq 1} \text{BPTIME}(n^c) \]
\[ RP = \bigcup_{c \geq 1} \text{RTIME}(n^c) \]
\[ ZPP = \bigcup_{c \geq 1} \text{ZPTIME}(n^c) \]

- BPP: two-sided bounded error, polynomial time
- RP: one-sided bounded error, polynomial time
- ZPP: zero-sided error, expected running time polynomial
Definition (\(\rho\)-Approximation algorithms)

Let \(\rho < 1\). A \(\rho\)-approximation algorithm \(A\) for an optimization problem returns for every input \(x \in \{0, 1\}^*\) a solution with quality at least \(\rho \cdot \text{val}(x)\), where \(\text{val}(x)\) denotes the quality of an optimal solution for \(x\).

Theorem (PCP)

There exists \(\rho < 1\) such that for every \(L \in \text{NP}\), there is a polynomial-time function \(f\) mapping strings to 3CNF formulas such that:

- if \(x \in L\), then \(\text{val}(f(x)) = 1\)
- if \(x \notin L\), then \(\text{val}(f(x)) < \rho\)

- If for every \(\rho < 1\) there is a polynomial-time \(\rho\)-approximation algorithm for MAX3SAT, then \(P = \text{NP}\).
We can solve 3SAT in time $2^{O(n)}$, where $n$ is the number of propositional variables in the input.

**Definition ($\delta_q$)**

For $q \geq 3$, let $\delta_q$ be the infimum of the set of constants $c$ for which there exists an algorithm solving $q$SAT in time $O(2^{cn}) \cdot m^{O(1)}$, where $n$ is the number of variables in the $q$SAT input and $m$ the number of clauses.

**Definition (ETH)**

*Exponential-Time Hypothesis* (conjecture): $\delta_3 > 0$.

- The ETH implies that there is no $2^{o(n)}$-time algorithm that solves 3SAT, and therefore also that $P \neq NP$. 
Average-case complexity

**Definition (Distributional problems)**

A *distributional problem* \( \langle L, D \rangle \) consists of a language \( L \subseteq \{0, 1\}^* \) and a sequence \( D = \{D_n\}_{n \in \mathbb{N}} \) of probability distributions, where each \( D_n \) is a probability distribution over \( \{0, 1\}^n \).

**Definition (The class distP)**

\( \langle L, D \rangle \) is in the class distP if there exists a TM \( M \) that decides \( L \) and a constant \( \epsilon > 0 \) such that for all \( n \in \mathbb{N} \):

\[
\mathbb{E}_{x \in \mathbb{R}D_n} \left[ \text{time}_M(x)^\epsilon \right] \text{ is } O(n).
\]

**Definition (The classes distNP and sampNP)**

distNP: all \( \langle L, D \rangle \) for which \( L \in \text{NP} \) and \( D \) is P-computable.

sampNP: all \( \langle L, D \rangle \) for which \( L \in \text{NP} \) and \( D \) is P-samplable.
Any further questions?