Exercise 1 (2pt). Let \( A \subseteq \{0, 1\}^* \) be an NP-complete language. Let \( p \) be a polynomial and let \( M_A \) be a polynomial-time Turing machine such that, for all \( x \in \{0, 1\}^* \):

\[
x \in A \text{ if and only if there exists some } u \in \{0, 1\}^{p(|x|)} \text{ such that } M_A(x, u) = 1.
\]

(a) Define the set \( B = \{ \langle x, z \rangle \mid \text{there exists } z' \in \{0, 1\}^* \text{ such that } |zz'| = p(|x|) \text{ and } M_A(x, zz') = 1 \} \).

Prove that \( B \) is in NP.

(b) Suppose that we have access to \( A \) as an oracle. Basically this means that we have a subroutine that, given a string \( y \), tells in a single step whether \( y \in A \). (See Definition 3.4 of Arora & Barak, 2009.) Construct a polynomial-time Turing machine \( M_{\text{search}} \) (with access to an \( A \)-oracle) that, given \( x \in \{0, 1\}^* \), if \( x \in A \) outputs a string \( u \) such that \( M_A(x, u) = 1 \) and if \( x \not\in A \) outputs 0. (Describe how \( M_{\text{search}} \) works at a high level.) Use (a).

- **Hint**: use the fact that \( A \) is NP-complete.

Exercise 2 (2pt). Let \( A \subseteq \{0, 1\}^* \) be a language. When a Turing machine \( M \) has access to an \( A \)-oracle, we write \( M^A \). We say that \( A \) is auto-reducible if there is a polynomial-time Turing machine \( M^A \) with oracle access to \( A \) such that for all \( x \in \{0, 1\}^* \):

\[
x \in A \text{ if and only if } M^A(x) = 1,
\]

with the special requirement that on input \( x \) the Turing machine \( M^A \) is not allowed to query the oracle \( A \) for \( x \).

Suppose that \( A \) is NP-complete. Prove that \( A \) is auto-reducible. Use Exercise 1.

Exercise 3 (3pt). Prove that \( \text{NTIME}(n) \neq \mathbb{P} \).

- **NTIME\((n)\)** can be characterized as the set of all decision problems that can be verified in linear time with a linear-size certificate. That is, \( A \in \text{NTIME}(n) \) if and only if there is a linear-time Turing machine \( M \) and a constant \( c \) such that for all \( x \in \{0, 1\}^* \) it holds that \( x \in A \) if and only if there exists some \( u \in \{0, 1\}^{c|x|} \) such that \( M(x, u) = 1 \). You are allowed to use this characterization of \( \text{NTIME}(n) \).

- **Hint**: Use the Nondeterministic Time Hierarchy Theorem.

Exercise 4 (3pt). In this exercise, we will construct a decision problem \( A \subseteq \{0\}^* \) that is not auto-reducible, using diagonalization. (For a definition of auto-reducibility, see the previous homework sheet.)

(a) Consider the function \( b : \mathbb{N} \rightarrow \mathbb{N} \) such that \( b(0) = 1 \) and for each \( n > 0 \) it holds that \( b(n) = 2^{b(n-1)} \).

Show that there exists some \( i_0 \) such that for all \( i \geq i_0 \) it holds that \( b(i) > b(i-1)^{i-1} \).

(b) Let \( M \) be a polynomial-time oracle Turing machine that—when given input \( x \in \{0\}^* \)—does not query \( x \) to the oracle. Show that there exists some \( i \) such that \( M = M_i \), and \( M_i^O \) runs in time at most \( n^i \) for all oracles \( O \).

- **Hint**: Remember that we can choose our representation scheme \( i \mapsto M_i \) in such a way that every Turing machine has infinitely many representations.
(c) Suppose that $M_i^O$—from (b)—is given the string $0^{b(i)}$ as input. What can you say about the size of the queries that $M_i^O$ makes to $O$?

(d) Construct a set $A \subseteq \{0\}^*$ that is not auto-reducible. Construct $A$ in stages $A_i$ such that $A = \bigcup_{i \geq 1} A_i$. Recursively define $A_i \subseteq \{0\}^{b(i)}$ in such a way that $A$ is not auto-reducible by construction. Make sure to prove that the set is not auto-reducible.

- *Hint:* suppose you have constructed $A_1, \ldots, A_{i-1}$. Let $A_{\leq i-1} = \bigcup_{1 \leq j \leq i-1} A_j$. Consider the behavior of machine $M_i^{A_{\leq i-1}}$ with oracle access to $A_{\leq i-1}$ when given input $0^{b(i)}$—that does not query $0^{b(i)}$. Based on the output of $M_i^{A_{\leq i-1}}$ on $0^{b(i)}$, choose whether $0^{b(i)}$ is in $A_i$ or not.