Exercise 1: Language theory
Show that the following problems are \textbf{PSPACE}-complete:

1. NFA Universality:
   - \textbf{INPUT}: a non-deterministic automaton \( A \) over alphabet \( \Sigma \)
   - \textbf{QUESTION}: \( L(A) = \Sigma^* \)?
   Bonus: what is the complexity of this problem for a DFA?

2. NFA Equivalence
   - \textbf{INPUT}: two non-deterministic automata \( A_1 \) and \( A_2 \) over the same alphabet \( \Sigma \)
   - \textbf{QUESTION}: \( L(A_1) = L(A_2) \)
   Bonus: what is the complexity of this problem for a DFA?

3. DFA Intersection Vacuity:
   - \textbf{INPUT}: deterministic automata \( A_1, \ldots, A_m \) for some \( m \)
   - \textbf{QUESTION}: \( \bigcap_{i=1}^m L(A_i) = \emptyset \)?

Exercise 2: Did you get padding?
Show that if \( P = \text{PSPACE} \), then \( \text{EXPTIME} = \text{EXPSPACE} \).

Exercise 3: Too fast!
Show that \( \text{ATIME}(\log n) \neq L \).

Exercise 4: Direct application
Show that \( \text{EXPSPACE} = \text{AEXPTIME} \).

\textit{Hint:} You may use that if \( f \) is space-constructible, then:

\[ \text{SPACE}(\text{poly}(f(n))) = \text{ATIME}(\text{poly}(f(n))) \]

Exercise 5: Closure under morphisms
Given a finite alphabet \( \Sigma \), a function \( f : \Sigma^* \to \Sigma^* \) is a morphism if \( f(\Sigma) \subseteq \Sigma \) and for all \( a = a_1 \cdots a_n \in \Sigma^* \), \( f(a) = f(a_1) \cdots f(a_n) \) (\( f \) is uniquely determined by the value it takes on \( \Sigma \)).

1. Show that \( \text{NP} \) is closed under morphisms, that is: for any language \( L \in \text{NP} \), and any morphism \( f \) on the alphabet of \( L \), \( f(L) \in \text{NP} \).

2. Show that if \( \text{P} \) is closed under morphisms, then \( \text{P} = \text{NP} \).

Exercise 6: Unary Languages
Prove that if a unary language is \( \text{NP} \)-complete, then \( \text{P} = \text{NP} \).
\textit{Hint:} consider a reduction from \text{SAT} to this unary language and exhibit a polynomial time recursive algorithm for \text{SAT}

2. Prove that if every unary language in \( \text{NP} \) is actually in \( \text{P} \), then \( \text{EXP} = \text{NEXP} \).