Exercise 1. \hspace{1cm} \text{PSPACE-complete}

Show that $\text{SPACE-TMSAT} = \{ \langle M, w, 1^n \rangle \mid \text{DTM } M \text{ accepts } w \text{ in space } n \}$ is PSPACE-complete.

Exercise 2. \hspace{1cm} \text{Hierarchy!}

For every space constructible $S : \mathbb{N} \to \mathbb{N}$, show that
\[\text{DTIME}(S(n)) \subseteq \text{SPACE}(S(n)) \subseteq \text{NSPACE}(S(n)) \subseteq \text{DTIME}(2^{O(S(n))})\]

Exercise 3. \hspace{1cm} L, NL

1. Show that $\text{EVEN} := \{ x : x \text{ has an even number of 1s} \}$ is in L.
2. Show that the language of balanced parentheses (with only one kind of parenthesis) is in L.
3. Show that $\text{SUM} := \{ \langle a, b, a+b \rangle \mid a, b \in \mathbb{Z} \}$ is in L.
4. Does this adapt to multiplication?
5. Show that $\text{PATH} := \{ \langle G = (V, E), x \in V, y \in V \rangle \mid \text{There exists a path between } x \text{ and } y \text{ in } G \}$ is in NL.

Exercise 4. \hspace{1cm} \text{Subtle space}

Let $f, g : \Sigma^* \to \Sigma^*$ such that $f$ is computable in polynomial space and $g$ is computable in logarithmic space.

1. Show that $f \circ g$ and $g \circ f$ are computable in polynomial space.
2. Does this still hold if $g$ is computable in polynomial space?

Exercise 5. \hspace{1cm} \text{Fast and Furious}

In this exercise, we focus on the fact that time and space complexities are defined up to a constant.

1. Let $c > 0$ a fixed integer, and $M$ be a Turing machine working on alphabet $\Gamma$. Show how a TM $M'$ can simulate $c$ steps of $M$ in a constant number of steps (6 is enough).
2. Use the previous question to show that for all $\epsilon > 0$, if a language $L$ is recognized by $M$ in time $\leq T(n)$, then there is a TM $M'$ recognizing $L$ in time $\leq (1 + \epsilon)n + \epsilon T(n)$
3. Use the previous question to show that for all $\epsilon > 0$, if a language $L$ is recognized by $M$ in space $\leq S(n)$, then there is a TM $M'$ recognizing $L$ in space $\leq \epsilon S(n)$.
Exercise 6.

We are going to show that a language is regular if and only if it needs \( o(\log \log(n)) \) space.

1. Optimality

   Besides, let us define \( W = \{ w_k; k \geq 0 \} \), where \( w_k \) is the concatenation of all length-\( k \) strings in lexicographical order, separated by a new symbol \( # \):

   \[
   w_k = 0^{k-2}00#0^{k-2}01#0^{k-2}10#0^{k-2}11# \cdots #1^k
   \]

   1. Show that \( W \) is not a regular language.
   2. Show that \( W \in \text{DSPACE}(\log \log(n)) \).
   3. What does it imply?

2. Simpler Implication

   4. Show that a language is regular if and only if it is recognized in constant (work-) space by a Turing Machine with write only output tape, and read only input tape with input head only going from left to right.
   5. What lemma is needed to deduce the easy implication of the exercise’s goal? Admit it for the moment (and prove it if you get bored after finishing everything).

3. Hard Work

   Suppose now the language is not regular. Let \( M \) be a Turing machine (on alphabet \( \Sigma \)) which decides it. We want to exhibit a constant \( c > 0 \) such that for infinitely many \( n \)'s, there is an input \( x \) with \( |x| = n \), on which the computation \( M(x) \) uses \( \geq c \log \log n \) (working) space.

   6. Show that for any \( k \), there is an input \( x \) such that \( M(x) \) takes \( \geq k \) space to compute.
   7. Pick now \( k_1 (k_1 = 100) \), and be \( x_1 \) of minimal length requiring \( s_1 \geq k_1 \) space to compute on.

      (a) Recall from HW 1 what a crossing sequence is. Define now a snapshot as a configuration of the machine. Compute the maximal number \( N \) of possible snapshots there can be during the computation of \( M(x) \)? During this execution, how many crossing sequences are there of length \( \leq m \)? \((m \geq 0)\)

      (b) Show that no two crossing sequences of \( M(x_1) \) (for two distinct indices) can be equal. What bound relating \( n_1 \) and \( N \) can you infer?

      (c) Let \( m \) is the maximal length of a crossing sequence for \( M(x_1) \). Show that \( m \leq N \). Compute a constant \( c \) answering the question for a length \( k_1 \).

   8. Conclude.