Lab problems are an opportunity for discussion and trying many different solutions. They are **not counted towards your grade**, and **you do not have to submit your solutions**.

The purpose of these problems is to get more comfortable with reasoning and writing about Turing machines. You should be **practicing writing out descriptions and proofs** for your solutions to these problems. If you are stumped or looking for guidance, some of these problems are in the "selected solutions" portion of the textbook.

1. Give an **implementation-level** description of a Turing machine that decides the following language over the alphabet \( \{0, 1\} \):

\[ \{w \mid w \text{ contains twice as many 0s as 1s}\} \]

2. Show that \( \text{ALL}_{\text{DFA}} \) is decidable, where \( \text{ALL}_{\text{DFA}} \) is defined as:

\[ \text{ALL}_{\text{DFA}} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) = \Sigma^*\} \]

3. Show that \( \text{INFINITE}_{\text{DFA}} \) is decidable, where \( \text{INFINITE}_{\text{DFA}} \) is defined as:

\[ \text{INFINITE}_{\text{DFA}} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) \text{ is an infinite language}\} \]

4. In lecture we saw that if an enumerator \( E \) enumerates a language \( A \) then there is a Turing Machine that recognizes \( A \).

Define a **shortlex enumerator** to be a Turing Machine with a printer that prints out strings in shortlex (i.e., short lexicographical) order. Suppose a shortlex enumerator enumerates a language \( A \). Show that \( A \) is decidable by giving a **high-level description** of a Turing Machine that decides \( A \).