Problem 1

\[ L = \{ \langle M \rangle | M \text{ accepts } \emptyset \} \]

- Is \( L \) recognizable?
  It is not recognizable. The intuition here is that if we are trying to check if the machine accepts nothing, we can wait and wait and still not know; if we wait just a little bit longer, the machine might accept.

- Is \( \bar{L} \) recognizable?
  This of course is recognizable. However, we have to be careful with the way we build the recognizer to prove this!

The dovetailing process can be used to build a recognizer for a language as follows:

  i. Begin running the machine on 1 string for 1 step
  ii. Generate 1 more string and run machine 1 step on that string and 2 steps on the previous string.
  iii. Generate the 3\textsuperscript{rd} string. Run 1 step on the third string, and an extra step on all the previous strings
  iv. Then add the 4\textsuperscript{th} string in. Run them all at the same time, running one step extra for each of the previous strings generated.

Essentially, we are executing all strings in parallel, adding new strings as we go. Sooner or later, we will execute the string that gets accepted, if there is one, enough steps to get accepted.

As soon as we find one accepting string, we know that \( \langle M \rangle \) is in \( \bar{L} \). \( \bar{L} \) is thus recognizable which implies that \( L \) must be unrecognizable since it is also undecidable.
Problem 2

\[ L = \{(M) | M \text{ accepts one of the following strings } "101", "001", "111" \} \]

• Is \( L \) recognizable?
  Yes! The dovetailing approach works to build a recognizer for \( L \! \).

• Is \( \overline{L} \) recognizable?
  No. \( L \) is undecidable (see below) and recognizable, so \( \overline{L} \) is unrecognizable.

The undecidability of \( L \) is proven via reduction to \( A_{101} \), which we know is undecidable. We begin by assuming that a decider for \( L, D_L \), exists. We will then use it to build a decider for \( A_{101} \). The decision procedure for \( A_{101} \) is as follows:

– Alter \( M \) to produce a new machine \( M' \) that immediately rejects “001” and “111” and then defers control to the original \( M \).
– Run \( D_L \) on \( M' \)
– If \( D_L \) accepts, then ACCEPT
– If \( D_L \) rejects, then REJECT

Case 1: \( M \in A_{101} \)
Since we know that \( M' \) will immediately reject “001” and “111”, if \( D_L \) accepts it must be because \( M \) accepts 101. Is this case our decider for \( A_{101} \) will also accept.

Case 1: \( M \notin A_{101} \)
If \( M \) does not accept 101, then \( D_L \) will reject \( M' \), causing our \( A_{101} \) to correctly reject.