Problem 1

Prove that the following language is undecidable:

\[ L = \{ \langle M, w \rangle \mid M \text{ is a TM that rejects input } w \} \]

We will prove that \( L \) is undecidable by reduction to \( A_{TM} \):

Suppose, for the sake of contradiction, that \( L \) is decidable. Then some machine \( D_L \) decides it. We use \( D_L \) to construct a decider for \( A_{TM} \).

\( D_{ATM} \) would proceed as follows on input \( \langle M, w \rangle \):

1. Use \( M \) and \( w \) to construct \( M' \) (see below)
2. Run \( D_L \) on \( \langle M', 1 \rangle \)
   - if \( D_L \) accepts, ACCEPT
   - if \( D_L \) rejects, REJECT

\( M' \) on input \( x \):

1. On input 1, run \( M \) on \( w \)
   - if \( M \) accepts, REJECT
   - if \( M \) rejects, LOOP

Case 1: \( \langle M \rangle \in A_{TM} \) In this case \( M \) accepts \( w \). This means that \( M' \) will reject, causing \( D(\langle M', 1 \rangle) \) to accept, followed by \( D_{ATM} \).

Case 2: \( \langle M \rangle \notin A_{TM} \) In this case \( M \) either rejects or loops on input \( w \). In both of these scenarios, \( M' \) will infinitely loop and never reject. Thus \( D(\langle M', 1 \rangle) \) will reject, followed by \( D_{ATM} \). Since we know that \( A_{TM} \) is undecidable it follows that \( L \) must also be undecidable.
Problem 2

Prove that the following language is unrecognizable:

\[ L = \{ \langle M \rangle \mid M \text{ is a TM that accepts any input of the form } 0^n1^n \text{ for } n > 0 \} \]

Intuition: First and foremost, since \( L \) depends on a non-trivial property of the language that a TM decides, we know by Rice’s Theorem that it is undecidable. From there, since a recognizer for \( L, R_L \), would have to verify acceptance on an infinite number of strings, there is no way for \( R_L \) to accept in a finite amount of time. Thus \( L \) is unrecognizable.

Proof: We will prove \( L \) is unrecognizable by showing that \( \overline{A_{TM}} \leq_m L \).

The reducing function works as follows:

On input \( \langle M, w \rangle \) where \( M \) is a TM and \( w \) is a string:

1. Construct a machine \( M' \) on input \( x \) that does the following:
   i. Run \( M \) on \( w \) for \( x \) steps
      If \( M \) hasn’t accepted after \( x \) steps, ACCEPT
      If \( M \) accepts within \( x \) steps, REJECT

2. Output \( \langle M', x \rangle \)

Case 1: \( \langle M, w \rangle \in \overline{A_{TM}} \) In this case \( M \) does not accept \( w \), so \( M' \) will accept on any input \( x \), ensuring that \( M' \in L \).

Case 2: \( \langle M, w \rangle \notin \overline{A_{TM}} \) In this case \( M \) accepts \( w \) at some step \( k \). So \( M' \) rejects any value of \( x \geq k \). Since it will always be possible to construct an input \( x \) of the form \( 0^n1^n \) that is greater than \( k \), then \( M' \) will not accept some input of the form \( 0^n1^n \). Thus, \( M' \notin L \).

BUT...ALWAYS MAKE SURE YOU’RE USING THE CORRECT COMPLIMENT. Consider the following doppelganger:

\( \overline{L}' = \{ \langle M \rangle \mid M \text{ is a TM that rejects some input of the form } 0^n1^n \text{ for } n > 0 \} \) is recognizable:

We can construct a recognizer for \( \overline{L}' \) by constructing a two-step TM. The first step will generate the next string of the form \( 0^n1^n, w_i \), and the second step will simulate \( M \) on \( w_0 \) through \( w_i \) using the dovetailing approach.

A recognizer for \( \overline{L}' \):

1. Create a counter \( i = 0 \)

2. Generate the \( i \)'th string of the language \( 0^n1^n \) by appending \( i \) 1’s to \( i \) 0’s

3. Simulate \( M \) on the \( i \)'th string for one step
4. Simulate $M$ on the $i-1$'th string for two steps

5. ...

6. Simulate $M$ on the 2nd string for $i-1$ steps

7. Simulate $M$ on the 1st string for $i$ steps

8. If any of these strings reject, ACCEPT

9. Increment $i$ and return to Step 2

Case 1: $\langle M \rangle \in \bar{L}'$ Our string enumerator will generate every possible string of the form $0^n1^n$. If $M$ rejects one of these strings at some point in its computation then $M$ is in $\bar{L}'$ and our recognizer will accept. We are guaranteed to reach the halting point if it exists via dovetailing.

Case 2: $\langle M \rangle \notin \bar{L}'$ If $M$ accepts all of these strings then $M$ is in not in $\bar{L}'$. In this case, the dovetailing will continue infinitely and our recognizer will never accept.

Therefore $\bar{L}'$ is recognizable. Since it is also undecidable, then $L$ must be unrecognizable. BUT NOT REALLY BECAUSE $\bar{L}'$ IS NOT THE CORRECT COMPLIMENT OF $L$. 