Consider the language \( 1_{TM} = \{ \langle M \rangle \mid |L(M)| = 1 \} \).

**Problem 1.** Identify what is wrong with the following proof that \( 1_{TM} \in \text{co–RE} \) and explain why.

To show that \( 1_{TM} \in \text{co–RE} \), we need to show that \( \overline{1_{TM}} \in \text{RE} \). To do that, we need only show that there is a TM \( M \) that recognizes it. To convenience, let \( \omega_1, \omega_2, \ldots \) be the lexicographical (and thus computable) enumeration of the input alphabet \( \Sigma^* \).

**Algorithm 1:** \( M \) = On input \( \langle N \rangle \),

\[
\text{for } i = 1 \text{ to } \infty \text{ do}
\]

\[
\text{for } j = 1 \text{ to } i \text{ do}
\]

\[
\text{Run } N(\omega_j) \text{ for } i \text{ steps}
\]

\[
\text{if } N \text{ accepted at least two distinct strings then}
\]

\[
\text{Accept}
\]

If \( N \) ever accepts a string, then it must do so in a finite amount of time. If \( N \) accepts two distinct strings, then it must do so in times \( t_1, t_2 < \infty \). Then when \( i \geq \max(t_1, t_2) \), \( N \) will accept both strings, and \( M \) will accept. This implies \( \langle N \rangle \in L(M) \).

On the other hand, when \( N \) does not accept more than one string, \( M \) clearly cannot accept. This implies \( \langle N \rangle \notin L(M) \).

Thus \( L(M) = \overline{1_{TM}} \).

**Problem 2.** Show that \( 1_{TM} \notin \text{DEC} \) by (Turing) reducing \( A_{TM} \) to it.