Problem 0. Document how much time you spend on each of the following problems and cite any resources you received help from.

Problem 1. When refactoring code, it is desirable to check if a line of code is never executed before deleting it. We sometimes call code that is unreachable *dead code*. For Turing machines, "dead code" is analogous to a machine never entering a certain state q on any input.

Prove or disprove that

$$\text{DEAD} = \{\langle M, q\rangle \mid M \text{ is a TM and } q \text{ is a state of } M \text{ that is never used}\}$$

is decidable.

Problem 2. The *hello world* language is the set of two strings

$$\text{HW} = \{\texttt{hello}, \texttt{world}\}.$$

Prove or disprove the following statement: A language L is decidable if and only if $L \leq_m \text{HW}$.

Bonus Problem. Recall that Turing machines can compute functions by writing output on their tape before halting and that functions are called *computable* if there is a Turing machine that computes them. Similar to how many languages are undecidable or unrecognizable, many functions are *uncomputable*.

For this problem, let $\Gamma = \{0, 1, \sqcup\}$ be the tape alphabet for all TMs. Now for $k \in \mathbb{N}$, let H_k be the set of all TMs that have exactly k states and halt on ϵ. For each $\langle M\rangle \in H_k$, let $\#1(M)$ be the number of 1s left on M's tape after running to completion on ϵ.

Define the function $f : \mathbb{N} \to \mathbb{N}$ as

$$f(k) = \max\{\#1(M) \mid \langle M\rangle \in H_k\}.$$

In other words, $f(k)$ is the maximum number of 1s any k-state TM can possibly write to its tape and not get into an infinite loop.

Prove that f is not a computable function.