Problem 1 (30 points) It is known that 3-COLORING is NP-complete. Show that 6-COLORING is NP-complete. (You do not need to show that it is in NP.)

Ans: We reduce 3-COLORING to 6-COLORING (the problem of asking if a graph can be colored by 6 or fewer colors such that no adjacent nodes have the same color). Given a graph $G(V, E)$ for 3-COLORING, the reduction outputs a graph $G'(V', E')$ by adding 3 new nodes with edges between each of the 3 nodes and all the other nodes in $V$. That is, $V' = V \cup \{x_1, x_2, x_3\}$ and $E' = E \cup \{(x_i, v) | v \in V', i = 1, 2, 3, x_i \neq v\}$. If $G \in 3$-COLORING, then $G' \in 6$-COLORING because 3 or fewer colors for the nodes in $V$ and additional 3 colors for those in $\{x_1, x_2, x_3\}$ suffice to make no adjacent nodes have the same color. Conversely, consider a legal coloring of $G'$ with 6 or fewer colors. In such a coloring, $\{x_1, x_2, x_3\}$ use up exactly 3 colors, leaving at most 3 colors for the nodes in $V$.

Problem 2 (30 points) Let $A \to B$ denote the set of functions from set $A$ to set $B$. (a) [15 points] How many functions in $\{0,1,2,3\}^n \to \{0,1\}$ are there? (b) [15 points] How many functions in $(\{0,1,2,3\}^n \to \{0,1\}) \to \{0,1,2\}^m$ are there? (Do not write something like $x^a b$ as it is ambiguous. Write $x^{(a^b)}$ or $(x^a)^b$.)

Ans: (a) $2^{(4^n)}$, (b) $(3^m)^{(2^{(4^n)})}$.

Problem 3 (15 points) Show that if $L$ and $\overline{L}$ are recursively enumerable, then $L$ is recursive.
Ans: Suppose that \( L \) and \( \overline{L} \) are accepted by the one-string Turing machines \( M \) and \( \overline{M} \), respectively. Then \( L \) is decided by a two-string Turing machine \( M' \), defined as follows. On input \( x \), \( M' \) simulates, on two different strings, \( M \) and \( \overline{M} \) on \( x \) in an interleaved fashion. That is, it simulates a step of \( M \), then a step of \( \overline{M} \), then again another step of \( M \), and so on. Since \( x \) is a member of \( L \) or \( \overline{L} \) (but not both), exactly one of the two machines will halt and accept. If \( M \) accepts, then \( M' \) halts on state “yes.” If \( \overline{M} \) accepts, then \( M' \) halts on “no.”

Problem 4 (25 points) Let \( L \) denote the language \( \{ M : M \text{ halts on all inputs} \} \). Show that \( L \) is not a recursive language, that is, membership in \( L \) is undecidable.

Ans: We know the Halting Problem \( H = \{ M; x : M(x) \neq \rangle \} \) is undecidable. Given the question “\( M; x \in H? \)”, we construct the machine: \( M_x(y) : M(x) \). \( M_x \) halts on all inputs if and only if \( M \) halts on \( x \). In other words, \( M_x \in L \) if and only if \( M; x \in H \). So if \( L \) were recursive, \( H \) would be recursive, a contradiction.