1. (10 pts) Let $A$, $B$, and $C$ be three strings each $n$ characters long. We want to compute the *longest subsequence* that is common to *all* three strings. Give a polynomial time algorithm for this problem. Derive the dynamic programming recurrence, clearly explain what each term stands for, and justify the correctness of the recurrence. Analyze the worst-case time complexity of your algorithm.

2. (10 pts) Recall that a clique is a collection of mutually adjacent vertices in a graph. The decision version of the CLIQUE problem takes as input a graph $G$ and an integer $k$, and decides if $G$ has a clique of size $k$ or not. The optimization version of the problem takes as input a graph $G$ and returns the largest clique in $G$. Show that the CLIQUE problem is self-reducible: that is, if the decision version has a polynomial time algorithm, then the optimization version also has a polynomial time algorithm.

3. (10 pts) Let $U$ be a finite set, and let $S = \{S_1, S_2, \ldots, S_m\}$ be a collection of subsets of $U$. Given an integer $k$, the SET-COVER problem asks if there is a sub-collection of $k$ sets $S' \subset S$ whose union covers all the elements of $U$. That is, $|S'| = k$, and $\bigcup_{S_i \in S'} S_i = U$. Prove that SET-COVER is NP-complete.

4. (20 pts) Suppose you are managing a communication network, modeled as a directed graph $G = (V, E)$. There are $m$ users and each user $i$ wants to reserve a path $P_i$ in the network to transmit his data. A set of path can be feasibly reserved if no two of them share any node in common.

The *Path Selection Problem* is the following: given a directed graph $G = (V, E)$, a set of path requests $P_1, P_2, \ldots, P_m$, and a positive integer $k$, is it possible to feasibly reserve at least $k$ of these paths?

Prove that the Path Selection Problem is NP-complete.