1. (20%) A string is a sequence of symbols; for example, $X = <x_1, x_2, ..., x_m>$ is a string of $m$ symbols $x_1, x_2, ..., x_m$. When we delete 0 or more symbols (not necessarily consecutive) from $X$, we get a subsequence of $X$. Write a dynamic programming algorithm to calculate the length of the longest common subsequence of $X = <x_1, x_2, ..., x_m>$ and $Y = <y_1, y_2, ..., y_n>$.

2. (20%) Write an algorithm to have the following input:
a connected weighted digraph $G(V, E)$, $w$, $x$, and $y$, where $V$ is the node set, $E$ is the edge set, the weight of the edge $(u, v)$ is stored in $w[u][v]$, and $x$ and $y$ are two nodes in $V$,
and the following output:
the shortest path from $x$ to $y$.

3. (10%) How can we prove that a given problem H is NP-hard?

4. (25%) Consider a general version of coin changing problem: given a set of coins of different denominations with unlimited quantities, you are asked to make changes for $K$ cents using these coins. Design an efficient algorithm to count the number of different ways of making changes. That is, if there are $n$ kinds of coins of denominations $c_1$, $c_2$, ..., $c_n$, the problem is equivalent to asking you to design an algorithm to count the number of different solutions for the following integer linear programming problem: Given positive integers $c_1$, $c_2$, ..., $c_n$, and $K$, find non-negative integer solutions $(x_1, x_2, ..., x_n)$ such that $x_1c_1 + x_2c_2 + ... + x_nc_n = K$. For example, if $c_1 = 1$, $c_2 = 5$, $c_3 = 10$, and $c_4 = 25$, then there are four ways of making changes for 11 cents and there are 13 ways of making changes for 26 cents.

5. (25%) Given a list of $n$ positive integers, $d_1$, $d_2$, ..., $d_n$, we want to efficiently determine whether there exists an undirected graph whose nodes have degrees precisely $d_1$, $d_2$, ..., $d_n$. This graph should not contain self-loops (edges with both endpoints equal to the same node) or multiple edges between the same pair of nodes. Design an algorithm to solve this problem.