1. A longest common subsequence of a set of strings \( \{ A_i \} \) is a longest string that is a subsequence of \( A_i \) for each \( i \). For example, \texttt{algorithm} and \texttt{altruistic}.

Given two strings \( A[1..n] \) and \( B[1..n] \), describe and analyze a fast dynamic programming algorithm that computes the length of a longest common subsequence of the two strings.

2. Describe and analyze a fast dynamic programming algorithm that computes the length of a longest common subsequence of three strings \( A[1..n] \), \( B[1..n] \), and \( C[1..n] \).

3. A lucky-10 number is a string \( D[1..n] \) of digits from 1 to 9 (no zeros), such that the \( i \)-th digit and the last \( i \)-th digit sum up to 10; in another words, \( D[i] + D[n - i + 1] = 10 \) for all \( i \). For example, \( 3141592648159697 \) and \( 11599 \) are both lucky-10 numbers. Given a string of digits \( D[1..n] \), describe and analyze a dynamic programming algorithm that computes the length of a longest lucky-10 subsequence of the string. [Hint: Try to use your solution to problem 1 directly.]

4. To think about later: Can you solve problem 1 in \( O(n) \) space?