1. The edit distance between two strings a and b is defined as

$$d(i,j) = \begin{cases} \max(i,j) & if \min(i,j) = 0 \\ \min \begin{cases} d(i,j-1), \\ d(i-1,j), \\ d(i-1,j-1) + [a_i = b_i] \end{cases} & \text{otherwise} \end{cases}$$

where a_i indicates the ith character of a and $[a_i \neq b_i]$ is equal to 1 if $a_i \neq b_i$ and 0 otherwise. Write an efficient, memoized function for calculating the edit distance of two strings.

2. Given two strings $x = x_1x_2...x_n$ and $y = y_1y_2...y_m$, a *common substring* consists of positions i, j and a length k such that $x_ix_{i+1}...x_{i+k} = y_jy_{j+1}...y_{j+k}$. Let $LCS(i, j)$ denote the size of the longest common substring ending at positions i in x and j in y. Write a dynamic programming recursion for $LCS(i, j)$.

3. Given the recursion above, write pseudocode for finding the longest common substring of two strings.

4. Given two strings $x = x_1x_2...x_n$ and $y = y_1y_2...y_m$, a *common subsequence* consists of indices $i_1 < i_2 < ... < i_k$ and $j_1 < j_2 < ... < j_k$ such that $x_{i_1}x_{i_2}...x_{i_k} = y_{i_1}y_{i_2}...y_{i_k}$. Let $LCS(i,j)$ denote the size of the longest common subsequence contained within the strings $x_1x_2...x_i$ and $y_1y_2...y_j$. Write a dynamic programming recursion for $LCS(i,j)$.

5. Given the recursion above, write pseudocode for finding the longest common subsequence of two strings.

6. A *contiguous subsequence* of a list S is a subsequence made up of consecutive elements of S. For instance, if S is

$$5, 15, -30, 10, -5, 40, 10$$

then $15, -30, 10$ is a contiguous subsequence but $5, 15, 40$ is not. Let $D(i)$ denote the sum of the largest contiguous subsequence ending at position i. Write a dynamic programming recursion for $D(i)$.