1. Consider two strings $s[1..n]$ and $t[1..m]$, and let $d(i,j)$ be the edit distance between $s[1..i]$ and $t[1..j]$. Prove that $d(i+1,j+1) \in \{d(i,j), d(i,j)+1\}$.

2. Write the pseudocode of the $O(n+D^2)$ time algorithm for computing the edit distance.

3. To apply the four Russians trick for computing LCS we partition the whole dynamic programming table into blocks. If x is the value in the upper-left corner of the block, we subtract x from each value on the left and the upper boundary, and then consider the sequence of differences between the values on the left boundary and their upper neighbours and between the values on the upper boundary and their left neighbours. For LCS both sequences consist of zeroes and ones, so we can tabulate all possible sequences. What about the edit distance? Can you see what else can be observed there?

4. Assume that the only allowed operation is checking if $s[i] = t[j]$. Show that then we need $\Omega(nm)$ such operations to compute LCS of $s[1..n]$ and $t[1..m]$ in the worst case.

(3 points) 5. Given two strings $s[1..n]$ and $t[1..m]$, compute the edit distance between $s[1..n]$ and all cyclic rotations of $t[1..m]$ in $O(nm)$ time ($O(nm \log(n+m))$ is also acceptable).