1. Present the edit distance matrix $D(i,j)$ for strings $S_1 = aaabb$ and $S_2 = bbbaab$. What are the optimal alignments and the optimal edit transcripts between the strings?

2. We say that a string $C$ is a merge of strings $A$ and $B$, if it is obtained by merging the characters of $A$ and $B$ in their original relative order together. For example, string “KAAKELI” is a merge of strings “ALI” and “KAKE”. Present a dynamic programming algorithm that takes three strings $A[1 \ldots n]$, $B[1 \ldots m]$ and $C[1 \ldots n+m]$, and tests whether $C$ is a merge of $A$ and $B$. What is the complexity?

3. Let $S_1[1 \ldots n]$ and $S_2[1 \ldots m]$ be two strings with $n \leq m$. Their similarity (defined by a given scoring matrix $s$) can be computed by maintaining only $2n$ cells of the dynamic programming table. Present the algorithm. Could the similarity be computed efficiently using even less space?

4. (Gusfield, Ex. 11.13) In a dynamic programming table for edit distance, are the entries along a row necessarily nondecreasing? What about down a column or a diagonal of the table? How about in a dynamic programming table for string similarity? (Assume negative scores for mismatches and spaces, and positive scores for matches.)

5. Consider locating approximate occurrences of pattern $P =$“aino” in the text “vaitonainen”. Let us score matches by 1, and mismatches and insertions/deletions by $-1$. Present the dynamic programming table. Explain how the approximate occurrences of $P$ are found, if we require their similarity with the pattern to be at least 2.