1. **Levenshtein Distance**: Given two strings $A = a_1a_2 \ldots a_k$ and $B = b_1b_2 \ldots b_l$ made up of symbols from a common alphabet $X$, define the Levenshtein (or string-edit) distance $L(A, B)$ between them to be the minimum number of insertions, deletions and substitutions of letters required to transform $A$ into $B$.

   (a) Show that $L(\cdot, \cdot)$ is a bona fide distance. In other words, argue why
   
   i. $L(A, A) = 0$ for all strings $A$,
   
   ii. $L(A, B) = L(B, A)$ for all strings $A$ and $B$, and
   
   iii. $L(A, C) \leq L(A, B) + L(B, C)$ for all strings $A$, $B$ and $C$.

   (b) Let $X = \{\alpha, \beta, \gamma, \delta\}$, and consider designing, for each symbol $a \in X$, an elementary weighted finite state acceptor $F_a$ with unique start- and end-states, and arcs labeled $\langle x/c \rangle$ to denote the symbol $x \in X \cup \{\epsilon\}$ and cost $c \in \{0, 1\}$, where $\epsilon$ represents the null symbol. Design the elementary acceptors such that in the acceptor obtained by concatenating the elementary acceptors $F_{a_1} \circ F_{a_2} \circ \cdots \circ F_{a_k}$ of the symbols of $A$, the minimum cost of accepting $B$, from the start-state of $a_1$ to the end-state of $a_k$, is exactly the Levenshtein distance $L(A, B)$. Draw the elementary acceptor $\{F_a, a \in X\}$, and clearly label the symbol and cost on each arc.

   (c) Modify the Viterbi algorithm in the textbook to construct an algorithm that computes $L(A, B)$ from the concatenated acceptor for $A$ as described above.

2. **Dynamic Time Warping**: This is the popular name of a technique that was successfully used for small-vocabulary, isolated-word recognition for a considerable period of time. Many variants of DTW have been proposed, and the following is an illustrative example.

   For each word $v$ in the vocabulary $\mathcal{V}$, a template $B(v) = \langle b_1(v) \ b_2(v) \ \ldots \ b_{l(v)}(v) \rangle$ and, for each $b_j$, a cost $c(a|b_j)$ over the acoustic symbols $a \in A$ is provided. In order to recognize an utterance $A = a_1, a_2, \ldots, a_k$, its best cost

   $$C_A(v) = \min_{j_1, \ldots, j_k} \sum_{t=1}^{k} c(a_t | b_{j_t}), \quad 1 = j_1 \leq j_2 \leq \ldots \leq j_k = l(v),$$

   (1)

   is computed for each word in the vocabulary, and the word with the lowest cost is chosen. Some additional restrictions are (usually) placed on the possible sequences $j_1, \ldots, j_k$ over which the minimum is computed, such as $j_{t+1} \leq j_t + 2$ for all $t = 1, 2, \ldots, k - 1$. 

(a) Explain the name *dynamic time warping*. (Hint: set $k = 8$ and $l(v) = 5$, and pick any two sequences $j_1, \ldots, j_8$ that satisfy the conditions above. For each sequence, plot the points $\{(t, j_t), t = 1, \ldots, 8\}$ on a rectangular grid with $l(v)$ horizontal lines and $k$ vertical lines, and “connect the dots.” Examine the two resulting mappings from the interval $[1, k]$ to the interval $[1, l(v)]$.)

(b) Over roughly how many admissible sequences $j_1, \ldots, j_k$ does the minimum of (1) need to be evaluated?

(c) Design an algorithm to efficiently compute the minimum of (1). (Hint: In a manner similar to the Viterbi algorithm, think about two different sequence-prefixes $j_1, \ldots, j_l$ and $j'_1, \ldots, j'_l$ with $(l, j_l) = (l', j'_l)$ as two paths that end up at the same “grid node.” How do their completions $j_1, \ldots, j_l, j_{l+1}, \ldots, j_k$ and $j'_1, \ldots, j'_l, j'_{l+1}, \ldots, j_k$ compare in the minimization?)

(d) How is the DTW model connected with fenonic baseforms? In particular, can you suggest a procedure for estimating the costs $c(a|b_j)$ given the template and sample utterances of each word?

3. Work on Project #2.