Assignment 05: String Edit Distance and Dynamic Programming

1. **String Edit/Levenshtein Distance** Recall that the string edit distance $d_S(\mathbf{x}_1, \mathbf{x}_2)$ between two strings $\mathbf{x}_1$ and $\mathbf{x}_2$ was defined as the minimum number of insertions, deletions, and substitutions required in order to match the strings, i.e., to map one string to another. The computation of this string edit distance was facilitated by a dynamic programming array, whose $(m, n) - \text{th}$ entry $d(m, n)$ was defined as $d(m, n) =$ min. # of edits req. to match ${}_1^m\mathbf{x}_1$ and ${}_1^n\mathbf{x}_2$.

   (a) Complete the dynamic programming array $d(m, n)$ in Fig. 1 to obtain the string edit distance between the strings $\mathbf{x}_1 =$ "$arcade$" and $\mathbf{x}_2 =$ "$aradywe$". Indicate the string edit distance and, on the completed array, indicate the optimal traceback path for the alignment between the two strings.

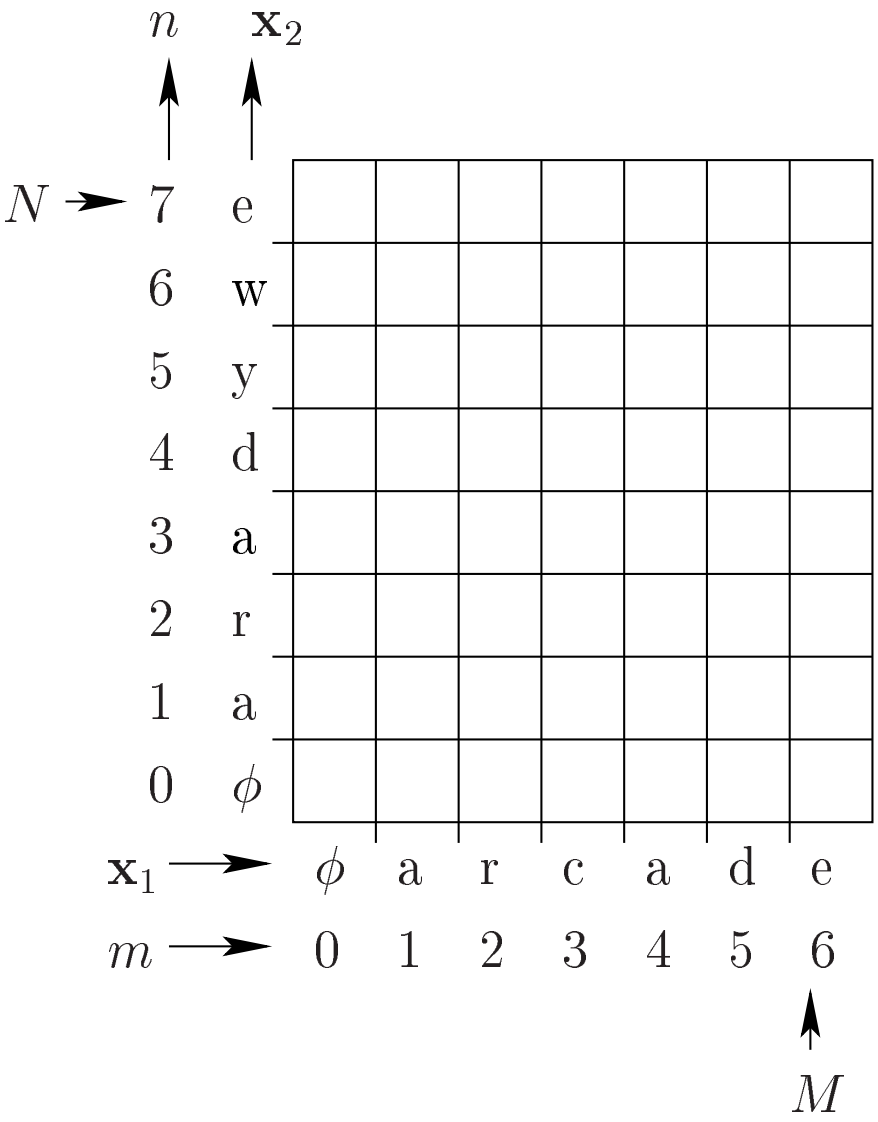

Figure 1: Blank array for computation of string edit distance for Problem 1a

   (b) Write a program for computing the string-edit distance between two strings (using dynamic programming) and use your program to enter in the string edit distances in Table 1.

$\mathbf{x}_1$	$\mathbf{x}_2$	$d_S(\mathbf{x}_1, \mathbf{x}_2)$
jaguar	jaduger	
hardhat	ardht	
jackals	jklsadue	
speedy	sperpyw	
sedative	sdatoive	

Table 1: String edit distances for pairs of strings