We transform a word (string of characters) into another word using the following “edit” operations: delete, insert, replace. For instance, here is how to turn “NAIVE” into “FANATIC:"

NAIVE - NAIVC - NAIC - NATIC - FNATIC - FANATIC.

The sequence of operations was REP, DEL, INS, INS, INS. The edit-distance of two words is the minimum number of edit operations needed to turn one word into the other. (If the above sequence of operations is optimal, then the edit-distance of NAIVE and FANATIC is 5.)

Describe an algorithm which finds the edit-distance of two given words in $O(km)$ steps where $k$ and $m$ are the respective lengths of the two input words.

Describe your algorithm in pseudocode. It should be very simple, no more than a few lines. Name the algorithmic technique used. Define the meaning of your variables. Half the credit goes for the clear definition (the “brain” of your algorithm).