For those who didn't get the solution, this may help

Let's consider 2 facts:
First:
If the character A[i] != B[j] and A[i-1] = B[j] , then A[i] MUST be moved to the beginning of the string to reach the case where the corresponding current characters in both strings are equal.

Example:
A = "BEC"
B = "BCE"
We can see that A[2] != B[2] AND A[1] = B[2], then we MUST move A[2] to the beginning of the string A to guarantee that the last characters in both strings are equal.
Generalize the above fact, to find the number of required operations.

Second:
If we moved a number of characters to the beginning, we can always get them in the order we want by choosing which will be moved first.

Eample:
lets consider the string "ABC"
if we wanted to reach "BCA", we have to move 'C' first, then 'B'
if we wanted to reach "CBA", we have to move 'B' first, then 'C'

Let's combine the 2 facts.
If we know which characters must be moved to the beginning of the strings, we will not have to do any additional operations since we can move them in a way to be in the order we want.