【算法思路和实现要点】

此题实际上是关于两个字符串的最长公共子序列（LCS）的，因为需要的操作次数为n+m-2LCS。

但是使用LCS动态规划算法会超时，因为n、m最大为500000.因此，考虑使用滚动数组缩减一维，同时利用超过k就返回-1来剪枝，即另一维只需要维护一个长度是2k+1的数组（不超过201，在当前动态规划位置的两侧各100个字符），从而也大幅缩减其数量级。

具体而言，dp数组的意义是对于A的前i个字符变换到B的前j个字符，需要的操作是dp[i][j].对于dp过程，其递归方程是dp[i][j]=min(dp[i-1][j]+1,dp[i][j-1]+1,dp[i-1][j-1])。为了减少数组的长度，i这一维使用滚动数组prev和curr表示，滚动n次；j这一维通过将角标变为j-i+k，同时将j在有意义的基础上控制在i-k和i+k之间，从而使得数组的总长度为2k+1.这是因为假如i和j的差超过了k，这时的计算是毫无意义的，因为至少要k+1步才能使A的前i个变为B的前j个，从而不可能在后续过程中小于k，因此不再继续计算。

【渐进时间复杂度的分析，包含过程】

本程序采取外层i从1到n循环，内层遍历数组内最多2k+1个元素的双循环，因此总体复杂度是O（nk）。

【渐进空间复杂度的分析，包含过程】

本程序使用了两个长度为2k+1的数组，因此总的空间复杂度是O（k）。

【遇到的困难和解决方法】

首先遇到了LCS动态规划算法的优化问题，在AI的提示下想到了可以放弃i-j大于k的情况优化了时间复杂度以及使用滚动数组，在答疑坊志愿者的提示下缩减了数组长度，优化了空间复杂度

【估计完成本题所用时间 】8小时

【关于本题的更多感想】无