Problem Statement **Suggest Edit**

Given two strings A and B consisting of lower case English letters. The task is to count the minimum number of pre-processing moves on the string A required to make it equal to string B after applying below operations:

```
1. Choose any index i (0 <= i < n) and swap characters a[i]  and b[i].
2. Choose any index i (0 <= i < n) and swap characters a[i]  and a[n-i-1] .
3. Choose any index i (0 <= i < n) and swap characters b[i]  and b[n-i-1] .
```

In one preprocess move, you can replace a character in A with any other character of the English alphabet.

Note:

```
The number of changes you make after the preprocess move does not matter.
You cannot apply to preprocess moves to the String B or make any preprocess moves
after the first change is made.
```

Input Format:

```
The first line of input contains an integer ‘T’ representing the number of test
cases. Then the test cases follow.

The first line of each test case contains string A consisting of lowercase English
letters.

The second line of each test case contains string B consisting of lowercase
English letters.
```

Output Format:

```
For each test case, print a single integer denoting the minimum number of pre-
processing moves on the string A required to make it equal to the string B. Return
-1, if it is impossible to make strings equal.

The output for each test case is in a separate line.
```

Note:

```
You do not need to print anything; it has already been taken care of.
```

Constraints:

```
1 <= T <= 100
1 <= N <= 5000

Where ‘T’ is the number of test cases, and ‘N’ is the length of the strings.

Time Limit: 1sec
```

Sample Input 1:

```
2
abacaba
bacabaa
zcabd
dbacz
```

Sample Output 1:

```
4
0
```

Explanation Of Sample Input 1:

```
For the first test case, preprocess moves are as follows: A[0] = ‘b’, A[2] = ‘c’, A[3] = ‘a’ and A[4] = ‘b’. Afterwards, A = ‘“bbcabba”. Then we can obtain equal strings by the following sequence of changes: swap(A[2], B[2]) and swap(A[2], A[6]). There is no way to use fewer than 4 preprocess moves before a sequence of changes to make strings equal, so the answer in this test case is 4.

For the second test case, no preprocess moves are required. We can use the following sequence of changes to make A and B equal: swap(B[1], B[5]), swap(A[2], A[4]).
```

Sample Input 2:

```
3
zxyyxzx
zyzyxyy
jfhjfl
jhkkjs
abcd
abcde
```

Sample Output 2:

```
3
4
-1
```