Solution description – Problem 3 - VIP

Student Posdarascu Eugenie Daniel – University of Bucharest

O(N*sigma) solution

Step 1:

First of all, we will try an approach by which we determine any correct solution, not necessarily the minimal lexicographic one. Moreover, we consider the special case N = K (all positions should differ).

If there is a character X whose sum of frequencies in the first, respectively the second string exceeds N, it is obvious from the principle of the box that we have no solution. Formally, noting f1[x] = frequency of character x in the first string and f2[x] = frequency of character x in the second string, if there is x for which f1[x] + f2[x] > N then we have no solution.
If there is no character x for which the above property is respected, then we can prove that there is always a solution. A possible approach for the reconstitution of the solution would be as follows
following:

1. Select the most frequent element from string 1, mark it with X.
2. Select the most frequent element from string 2, denote it with Y.
3. If X != Y, we can pair them. Otherwise, we pair X with Z, the second most common element of string 2.

Therefore, we can say that the invariant for this particular case is f1[x] + f2[x] <= N, for any X from 'a' to 'z'.

Step 2:

Now we will try to solve the problem for any K in the interval [0, N] (we still don't want the minimum lexicographic string).

Let's assume that from the N available positions, having K = the number of positions by which they differ the strings, we note P = N – K = the number of positions in which the strings are similar. We will try to fix all the P positions where the characters of the strings coincide in order to reduce the problem to the variant from step 1. Thus, an operation by which we select a character X from both string 1 and string 2, in order to pair them, the value of the sum f1[X] + f2[X] will decrease by two units. Since we want to reduce the problem to the particular case from step 1 in which the length has dimension K, the invariant becomes f1[X] + f2[X] <= K. Therefore, all the characters for which this property is not respected must paired to reduce the sum of their frequencies. In conclusion, an approach for this subproblem is as follows.

1. We select the character X with maximum f1[X] + f2[X].
2. Pair an X character from string 1 with an X character from string 2.
3. Repeat the procedure P times, if possible.
4. If we were able to update all the P pairings, and at the end f1[X] + f2[X] <= K for any X, we have a solution and we continue to solve by applying the procedure from step 1.

Thus, the invariant for this subproblem becomes:
1. For each character X where f1[X] + f2[X] > K, we calculate the number of operations necessary for this sum to decrease below K. More precisely, (f1[X] + f2[X] – k + 1)/2. Obviously, this value must be smaller than min(f1[X], f2[X]) (the maximum number of pairings we can make with the character X).

$$(f1[X] + f2[X] - k + 1) / 2 \le \min(f1[X], f2[X])$$

2. We note the total number of operations for all characters with total_op. This number must not exceed P (since we are allowed to apply a maximum of P pairings).

total_op = Sum((f1[X] + f2[X] – k + 1) / 2) <= P

3. The first two restrictions guarantee us that we do not need more than P pairings. However, we must apply fixed P pairings, so the maximum number of pairings we can do must be at least P.

P <= Sum(min(f1[X], f2[X]))

Step 3:

We will try to determine the minimal lexicographic solution. We select each position from the left to the right and try to fix each valid character in turn. If, after such a fixation, the invariant from step 2 is preserved, we have a solution. Otherwise, we try another character. Maintaining the invariant can be done both in O(1) and in O(sigma).

If the invariant from step 2 is not kept from the very beginning, the answer is -1.

Optimal final complexity: O(N * sigma). A careful implementation in O(N * sigma ^ 2) also gets 100 points.