

Analysis: Smaller Strings

Test Set 1

Let us calculate the upper bound of a brute-force solution in the worst case scenario.

Given the limits of Test Set 1, $1 \leq K \leq 5$ and $1 \leq N \leq 8$, we can compute the total number of strings, in the worst case scenario, to be $K^N = 5^8 = 390625$. This number is low enough to generate all of them, keeping track of the ones that are palindromes and lexicographically smaller than the input string S .

Test Set 2

For Test Set 2, the limits are much higher, so the strategy mentioned above is too slow.

Let us start by making an observation about the definition of lexicographically smaller strings. Say we have two strings A and B of the same length, and i is the first position at which A_i and B_i differ. By definition, if $A_i < B_i$, then A is lexicographically smaller than B . Now notice no matter what characters we append to the end of the strings A and B , we still have that A is lexicographically smaller than B .

Let us define $H = \lceil \frac{N}{2} \rceil$, which is the size of the first half of the input string S , rounded up. Also, let us define S' to be the prefix of length H from the input string S .

To come up with an efficient solution, we only need to count the number of strings T' of size H (not necessarily palindromes) are lexicographically smaller than S' (with the caveat below). The reason is based on the above observation.

Insight 1: If T' is lexicographically smaller than S' , then appending appropriate $N - H$ characters to make it a palindrome of size N , will keep it lexicographically smaller than S .

We can construct a palindrome string T of size N as follows. Let R' be the reverse of T' . If N is even, then combine T' and R' . For example, if $N = 6$ and $T' = "abc"$, then $T = "abccba"$. If N is odd, then combine T' (without the last character) and R' . And for example, if $N = 7$ and $T' = "abcd"$, then $T = "abcdcba"$.

Let us see how these ideas fit together with an example: Let us say that $S = "tomato"$ and $N = 6$. Then, we have that $H = \lceil \frac{6}{2} \rceil = 3$ and $S' = "tom"$. If we choose an arbitrary string $T' = "pea"$, we can observe that $"pea"$ is lexicographically smaller than $"tom"$. Therefore, we can build a palindrome string $T = "peaaep"$, which is guaranteed to be lexicographically smaller than $"tomato"$.

Hence, the problem can now be simplified to finding how many strings of size H (not necessarily palindromes) are lexicographically smaller than S' .

The only **caveat** is for the string $T' = S'$. By just analysing T' , we can not say whether T is going to be lexicographically smaller than S or not. Therefore, for this particular T' we should generate T manually and check if it is lexicographically smaller than S , and if so, add that to our answer.

Also notice we do not have to generate all of the possible lexicographically smaller strings and count them. Notice, if we assume first K letters of the alphabet as digits in K -based numeric

system, i.e. $a = 0, b = 1, \dots, z = 25$, then every string T' lexicographically smaller than S' can be mapped to a number that is smaller than the number corresponding to S' (and vice versa).

Insight 2: In order to calculate the number of strings lexicographically smaller than S' , we just need to convert the \mathbf{K} -based number corresponding S' to base 10.

For example, if $\mathbf{K} = 4$ and we want to find the number of strings lexicographically smaller than $S' = "da"$. Converting S' from base-4 to base-10 will give $3 \times 4^1 + 0 \times 4^0 = 12$ which is the number of strings smaller than $"da"$ which are $["aa", "ab", "ac", "ad", "ba", "bb", "bc", "bd", "ca", "cb", "cc", "cd"]$.