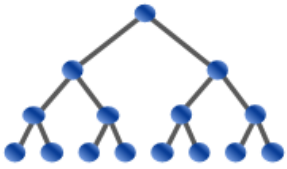

USACO 2024 US OPEN CONTEST, SILVER
PROBLEM 3. THE 'WINNING' GENE

Return to Problem List

Contest has ended.

Log in to allow submissions in analysis mode

English (en)

****Note: The memory limit for this problem is 512MB, twice the default.****

After years of hosting games and watching Bessie get first place over and over, Farmer John has realized that this can't be accidental. Instead, he concludes that Bessie must have winning coded into her DNA so he sets out to find this "winning" gene.

He devises a process to identify possible candidates for this "winning" gene. He takes Bessie's genome, which is a string S of length N where $1 \leq N \leq 3000$. He picks some pair (K, L) where $1 \leq L \leq K \leq N$ representing that the "winning" gene candidates will have length L and will be found within a larger K length substring. To identify the gene, he takes all K length substrings from S which we will call a k-mer. For a given k-mer, he takes all length L substrings, identifies the lexicographically minimal substring as a winning gene candidate (choosing the leftmost such substring if there is a tie), and then writes down the 0-indexed position p_i where that substring starts in S to a set P.

Since he hasn't picked K and L yet, he wants to know how many candidates there will be for every pair of (K, L).

For each v in $1 \ldots N$, help him determine the number of (K, L) pairs with $|P| = v$.

INPUT FORMAT (input arrives from the terminal / stdin):

N representing the length of the string. S representing the given string. All characters are guaranteed to be uppercase characters where $s_i \in A - Z$ since bovine genetics are far more advanced than ours.

OUTPUT FORMAT (print output to the terminal / stdout):

For each v in $1 \ldots N$, output the number of (K, L) pairs with $|P| = v$, with each number on a separate line.

SAMPLE INPUT:

```
8
AGTCAACG
```

SAMPLE OUTPUT:

```
11
10
5
4
2
2
1
1
```

In this test case, the third line of the output is 5 because we see that there are exactly 5 pairs of K and L that allow for three "winning" gene candidates. These candidates are (where p_i is 0-indexed):

```
(4, 2) -> P = [0, 3, 4]
(5, 3) -> P = [0, 3, 4]
(6, 4) -> P = [0, 3, 4]
(6, 5) -> P = [0, 1, 3]
(6, 6) -> P = [0, 1, 2]
```

To see how (4,2) leads to these results, we take all 4-mers

```
AGTC
GTCA
TCAA


CAAC
AACG
```

For each 4-mer, we identify the lexicographically minimal length 2 substring

```
AGTC -> AG
GTCA -> CA
```

TCAA -> AA
CAAC -> AA
AACG -> AA

We take the positions of all these substrings in the original string and add them to a set P to get $P = [0, 3, 4]$.

On the other hand, if we focus on the pair $(4, 1)$, we see that this only leads to 2 total "winning" gene candidates. If we take all 4-mers and identify the lexicographically minimum length 1 substring (using A and A' and A* to distinguish the different As), we get

AGTC -> A
GTCA' -> A'
TCA'A* -> A'
CA'A*C -> A'
A'A*CG -> A'

While both A' and A* are lexicographically minimal in the last 3 cases, the leftmost substring takes precedence so A' is counted as the only candidate in all of these cases. This means that $P = [0, 4]$.

SCORING:

- Inputs 2-4: $N \leq 100$
- Inputs 5-7: $N \leq 500$
- Inputs 8-16: No additional constraints.

Problem credits: Suhas Nagar

Contest has ended. No further submissions allowed.