Task: RLE
RLE Compression

Source file `rle.*`

Solution

The optimal solution should work in $O(n + m)$.

First we decode the input and store the sequence as a sequence of blocks. Each block represents a repetition of one character. Let the number of blocks be B. It satisfies $B \leq m$.

Straightforward dynamic solution leads to time and memory $O(nm)$. For each $i = 0, \ldots, B$ and each $e = 0, \ldots, n - 1$ we calculate a value $L(i, e)$ — the length of the shortest code of first i blocks such that at the end of the code the special character is set to e.

This algorithm can be accelerated using the following observations. First, the code of a block of repetition of a character a using the special character other than a is not longer than the code of the block using the special character $e = a$. Therefore, if $a \neq e$ it is always good choice to code the block using special character e, thus $L(i, e) = L(i - 1, e) + C$, where C is the length of the shortest code of block i given $e \neq a$. So all values of $L(i, e)$ for $e \neq a$ differ from $L(i - 1, e)$ by the same number C.

More attention is needed when calculating $L(i, a)$. We want to encode block i in a such way that after encoding it, the special character will be set to a. This can be done in two ways: with or without switching the special character. If we don't want to switch, then the length of the code will be equal to $L(i - 1, a)$ plus the length of the code of block i using a as the special character. If we want switch the special character to a somewhere, it is always worthwhile to do it at the end of block of a. This costs additional 3 characters. The rest of the code will contain C character plus the smallest value from the set $\{L(i - 1, b) \,|\, b \neq a\}$. To reconstruct later the code we need to save only how this particular value $L(i, a)$ was calculated.

We don't have to store $L(i, \cdot)$ for all i. We need only values $L(i, \cdot)$ for the current block.

There can be developed a fast data structure with the following operations running in constant time:

- getting the value $L(i, e)$ for any e,
- calculation of $L(i, \cdot)$ from $L(i - 1, \cdot)$,
- getting the smallest value $L(i, e)$ with $e \neq a$.

The key observation for doing it is that two values $L(i, \cdot)$ may differ at most by 3. This can be proved by a simple induction on the number of blocks.

The task require from a contestant some short insight into the problem. The more difficult part of the solution should be careful implementation.