Faster Elimination

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

This is the hard version of the problem. The only difference between the versions is the range of n.

Colin likes to play the Happy Eliminate game to kill time. Usually, the player is asked to find a fixed pattern in the given content, then eliminate it. The variety of patterns makes the game rich and exciting.

To research which pattern makes the game most exciting, Colin built a model: The contents given to the user are represented as a string s, and the pattern is another string t. In each turn of the game, the user can do one of the following two operations, and the game will be kept going until t becomes empty:

- Delete the last character of the pattern t.
- Find a substring of s that is equal to t, say $s[i, j] = t$, then delete several characters in the prefix of s until $s[i+1]$, i.e., (make s become $s[i+1, |s|]$), and then delete the last character of the pattern t.

Colin defines a game's excitement as the maximum number of times the second operation can be performed. He wants to find the best pattern to maximize excitement, but he has no idea how to construct it. Thus, he turned it over to you, the top player of Happy Eliminate.

Recall that $|s|$ represents the length of the string s, and $s[i, j]$ represents the substring of the string s, starting from the i-th character to the j-th character (including the i-th and the j-th).

Input

The first line contains a single integer T, representing the number of test cases.

For each test case:

The first line contains a single integer n ($1 \le n \le 5 \times 10^5$), representing the length of the string s.

The second line contains a string s, consisting only of lowercase letters.

It's guaranteed that the sum of n is not larger than 5×10^5.

Output

For each test case, output a single line with a single integer, representing the maximum possible excitement.

Example

standard input	standard output
3 9 abcabcabc 4 aaaa 5 ababa	3 4 3

Note

For the first sample, you can choose t as *abc*.

For the second sample, you can choose t as *aaaa*.

For the third sample, you can choose t as *aba*.