**Count the string**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 32768/32768 K (Java/Others)  
Total Submission(s): 7586    Accepted Submission(s): 3526**

Problem Description

It is well known that AekdyCoin is good at string problems as well as number theory problems. When given a string s, we can write down all the non-empty prefixes of this string. For example:  
s: "abab"  
The prefixes are: "a", "ab", "aba", "abab"  
For each prefix, we can count the times it matches in s. So we can see that prefix "a" matches twice, "ab" matches twice too, "aba" matches once, and "abab" matches once. Now you are asked to calculate the sum of the match times for all the prefixes. For "abab", it is 2 + 2 + 1 + 1 = 6.  
The answer may be very large, so output the answer mod 10007.

Input

The first line is a single integer T, indicating the number of test cases.  
For each case, the first line is an integer n (1 <= n <= 200000), which is the length of string s. A line follows giving the string s. The characters in the strings are all lower-case letters.

Output

For each case, output only one number: the sum of the match times for all the prefixes of s mod 10007.

Sample Input

1

4

abab

Sample Output

6

Author

foreverlin@HNU

分析：  
1 题目要求的是给定一个字符串s，求字符串s的所有的前缀在s的匹配的次数之和mod10007.  
2 很明显n<= 200000，分析一下那么就要n个前缀如果每一个前最都去匹配s的话复杂度就是o(n^2)，那么肯定是TLE的，所以要考虑另外的思路  
3 我们知道next[j] = len,表示的是在前j个字符里前缀和后缀的最大的匹配的长度为len，所以根据next数组的性质，我们只要去枚举j的值从n->1,为什么要从n开始而不是1开始呢，这里因为是要求前缀的匹配数而不是后缀；  
4 求sum的时候注意每一步都有可能超过范围，所以就要求一次sum同时取模一次。

#include <stdio.h>

#include <string.h>

#define MOD 10007

char a[200005],b[200005];

int next[200005];

int n,m,sum;

void Next() ///求next数组

{

next[0] = next[1] = 0;

for(int i = 1; i < n; i++)

{

int j = next[i];

while(j&&b[j]!=b[i])

j = next[j];

next[i+1] = b[i]==b[j]?j+1:0;

}

}

/\*

void find1() ///模式串与主串进行匹配

{

int j = 0; ///初始化在模式串的第一个位置

for(int i = 0; i < n; i++) ///遍历主串

{

while(j&&b[j]!=a[i])

j = next[j];

if(b[j]==a[i]) ///如果匹配成功则进行下一个位置

j++;

if(j==m) ///能在a数组中找到b数组

{

sum++;

j = 0;

}

}

}\*/

void find1()

{

sum = 0;

for(int i = n; i >= 1; i--)

{

sum = (sum + 1) % MOD;

int j = next[i];

while(j)

{

sum = (sum + 1)%MOD;

j = next[j];

}

}

printf("%d\n",sum);

}

int main()

{

int t;

scanf("%d",&t);

while(t--)

{

scanf("%d",&n);

scanf("%s" ,b);

Next();

find1();

}

return 0;

}