

String Similarity



n Submissions Leaderboard Discussions Editorial Tutorial
--

For two strings A and B, we define the similarity of the strings to be the length of the longest prefix common to both strings. For example, the similarity of strings "abc" and "abd" is 2, while the similarity of strings "aaa" and "aaab" is 3.

Calculate the sum of similarities of a string S with each of it's suffixes.

Input Format

The first line contains the number of test cases T. Each of the next T lines contains a string each.

Constraints

1 <= T <= 10

The length of each string is at most 100000 and contains only lower case characters.

Output Format

Output T lines containing the answer for the corresponding test case.

Sample Input

2 ababaa aa

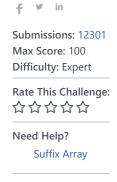
Sample Output

11 3

Explanation

For the first case, the suffixes of the string are "ababaa", "babaa", "babaa", "babaa", "abaa", "babaa". The similarities of these strings with the string "ababaa" are 6,0,3,0,1, & 1 respectively. Thus, the answer is 6+0+3+0+1+1=11.

For the second case, the answer is 2 + 1 = 3.



Z Function

LCP Array

More

```
Java 8
 Current Buffer (saved locally, editable) & 🗸
                                                                                                                              Ö
 1 ▼ import java.io.*;
 2 import java.util.*;
 3
 4 ▼ public class Solution {
 5
 6 1
        public static void main(String[] args) {
 7
 8 1
                     /*int L = 0, R = 0;
             for (int i = 1; i < n; i++) {
 9
10
               if (i > R) {
11
                 L = R = i;
                 while (R < n \&\& s[R-L] == s[R]) R++;
12
                 z[i] = R-L; R--;
13
14
               } else {
                 int k = i-L;
15
16
                 if (z[k] < R-i+1) z[i] = z[k];
17
                 else {
                   L = i;
18
19
                   while (R < n \&\& s[R-L] == s[R]) R++;
20
                   z[i] = R-L; R--;
21
22
              }
23
             }*/
24
25
             Scanner scan = new Scanner(System.in);
26
             int tst = scan.nextInt();
27
28 ▼
             for(int i = 0; i < tst; i++){
29
30
                 String str = scan.next();
31
                 int left = 0, right = 0;
32
                 long output = (long)str.length();
33
34
                 long[] Z = new long[str.length()];
35
                 for(int j = 1 ; j < str.length() ; j++){</pre>
36 ▼
37
                     if(j > right){
38 ▼
39
                         right = left = j;
40
41
                         while(right < str.length() && str.charAt(right) == str.charAt(right - left)){</pre>
42
43
                              right++;
44
45
                                output += (long)(right - left);
46
                                Z[j] = (long)(right - left);
                                right--;
47
48
                     }
49
                     else{
50
                         int a = j - left;
51
52
53 •
                         if((int)Z[a] < right - j + 1){
54 ▼
                            Z[j] = Z[a];
55 ₹
                              output+= Z[j];
56
                         }
57 ▼
                         else{
58
                              left = j;
59
60
61 ▼
                              while(right < str.length() && str.charAt(right) == str.charAt(right - left)){</pre>
62
                                  right++;
63
                              }
64
```

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature