String Similarity

by HackerRank

Problem | 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", "abaa", "baa", "aa" and "a". 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.

Submissions: 12301
Max Score: 100
Difficulty: Expert

Rate This Challenge:

Need Help?
Suffix Array

Z Function
LCP Array
More

Current Buffer (saved locally, editable) Java 8

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {

                /*int L = 0, R = 0;
        for (int i = 1; i < n; i++) {
          if (i > R) {
            L = R = i;
            while (R < n && s[R-L] == s[R]) R++;
            z[i] = R-L; R--;
          } else {
            int k = i-L;
            if (z[k] < R-i+1) z[i] = z[k];
            else {
              L = i;
              while (R < n && s[R-L] == s[R]) R++;
              z[i] = R-L; R--;
            }
          }
        }*/

        Scanner scan = new Scanner(System.in);
        int tst = scan.nextInt();

        for(int i = 0 ; i < tst ; i++){

            String str = scan.next();
            int left = 0, right = 0;

            long output = (long)str.length();
            long[] Z = new long[str.length()];

            for(int j = 1 ; j < str.length() ; j++){

                if(j > right){

                    right = left = j;

                    while(right < str.length() && str.charAt(right) == str.charAt(right - left)){
                        right++;
                    }
                          output += (long)(right - left);
                          Z[j] = (long)(right - left);
                          right--;
                }
                else{

                    int a = j - left;

                    if((int)Z[a] < right - j + 1){
                      Z[j] = Z[a];
                        output+= Z[j];
                    }
                    else{

                        left = j;

                        while(right < str.length() && str.charAt(right) == str.charAt(right - left)){
                            right++;
                        }


```

```
                    output += (long)(right - left);
                    Z[j] = (long) (right - left);
                    right--;
                }
            }
        }

        System.out.println(output);
    }
}
}
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code