Practice Compete Jobs Rank Leaderboard gadhiya

Dashboard > Algorithms > Strings > Ashton and String

Badge Progress ★ ★ ★ ★

Points: 4727.88 Rank: 491

Ashton and String

by amititkgp

Problem Submissions Leaderboard Discussions Editorial Tutorial

Ashton appeared for a job interview and is asked the following question. Arrange all the distinct substrings of a given string in lexicographical order and concatenate them. Print the K^{th} character of the concatenated string. It is assured that given value of K will be valid i.e. there will be a K^{th} character. Can you help Ashton out with this?

Note We have distinct substrings here, i.e. if string is `aa`, it's distinct substrings are `a` and `aa`.

Input Format

First line will contain a number T i.e. number of test cases.
First line of each test case will contain a string containing characters $(a - z)$ and second line will contain a number K.

Constraints

$1 \leq T \leq 5$
$1 \leq length \leq 10^5$
K will be an appropriate integer.

Output Format

Print K^{th} character (the string is 1 indexed)

Sample Input

```
1
dbac
3
```

Sample Output

```
c
```

Explanation

The substrings when arranged in lexicographic order are as follows

```
a, ac, b, ba, bac, c, d, db, dba, dbac
```

On concatenating them, we get

```
aacbbabaccddbdbadbac
```

The third character in this string is `c` and hence the answer.

Submissions: 2157

Max Score: 100

Difficulty: Advanced

Rate This Challenge:

☆☆☆☆☆

Need Help?

Suffix Array

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) {

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

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

            String str = scan.next();
            long k = scan.nextLong();

            String[] suffix = new String[str.length()];
            int[] lcp = new int[str.length()];

            for(int j = 0 ; j < str.length() ; j++){
                suffix[j] =  str.substring(j,str.length());
            }

            Arrays.sort(suffix);

            for(int a = 1 ; a < suffix.length - 1 ; a++){
                lcp[a] = lcpBST(suffix[a],suffix[a-1]);
            }

            long cnt = 0;

            boolean ans = false;

            for(int j = 0 ; j < suffix.length ; j++){

                long value = (((suffix[j].length() * (suffix[j].length() + 1)) / 2) - ((lcp[j] * (lcp[j] + 1)) / 2));

                if(cnt + value < k ){
                    cnt += value;
                }
                else{

                    for(int l = lcp[j] ; l < suffix[j].length() ; l++){

                        if(cnt + suffix[j].substring(0,l+1).length() >= k){
                            System.out.println(suffix[j].charAt((int)(k - cnt - 1)));
                            ans = true;
                            break;
                        }
                        else{
                            cnt += suffix[j].substring(0,l+1).length();
                        }
                    }
                }

                if(ans){
                    break;
                }

            }
        }
    }


    public static int lcpBST(String s1, String s2){

        int i = 0;
        int output = 0;

        while(i < s1.length() && i < s2.length()){

            if(s1.charAt(i) == s2.charAt(i)){
                output++;
                i++;
            }
            else{
                break;
            }
        }

        return output;

    }

}
```

Line: 1 Col: 1

Upload Code as File ☐ Test against custom input

Run Code Submit Code

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