ALL TRACKS > ALGORITHMS > STRING ALGORITHMS > > PROBLEM

Sonya and string shifts

Attempted by: **112** / Accuracy: **18%** / ★★★★☆

Tag(s): Medium-Hard

PROBLEM **EDITORIAL** **MY SUBMISSIONS** **ANALYTICS**

The problem "Sonya and string shifts " doesn't have any editorial. You can contribute it by sending editorial in markdown format to moderator@hackerearth.com.

Author Solution by Boris Sokolov

```
#include <bits/stdc++.h>
using namespace std;

const int INF = 2e9;
const int N = (int)1e6 + 9;
const int alphabet = 26;

char s[N];
int p[N], cnt[N], c[N], pn[N], cn[N], ans[N];
vector<int> v[N];

int main() {
    int n;
    scanf("%d\n", &n);
    gets(s);
    memset(cnt, 0, alphabet * sizeof(int));
    for (int i = 0; i < n; ++i) {
        ++cnt[s[i] - 'a'];
    }
    for (int i = 1; i < alphabet; ++i) {
        cnt[i] += cnt[i - 1];
    }
    for (int i = 0; i < n; ++i) {
        p[--cnt[s[i] - 'a']] = i;
    }
    c[p[0]] = 0;
    int classes = 1;
    for (int i = 1; i < n; ++i) {
        if (s[p[i]] != s[p[i - 1]]) {
            ++classes;
        }
        c[p[i]] = classes - 1;
    }
    for (int h = 0; (1 << h) < n; ++h) {
```

```
                for (int i = 0; i < n; ++i) {
                        pn[i] = p[i] - (1 << h);
                        if (pn[i] < 0) {
                                pn[i] += n;
                        }
                }
                memset(cnt, 0, classes * sizeof(int));
                for (int i = 0; i < n; ++i) {
                        ++cnt[c[pn[i]]];
                }
                for (int i = 1; i < classes; ++i) {
                        cnt[i] += cnt[i - 1];
                }
                for (int i = n - 1; i >= 0; --i) {
                        p[--cnt[c[pn[i]]]] = pn[i];
                }
                cn[p[0]] = 0;
                classes = 1;
                for (int i = 1; i < n; ++i) {
                        int mid1 = (p[i] + (1 << h)) % n;
                        int mid2 = (p[i - 1] + (1 << h)) % n;
                        if (c[p[i]] != c[p[i - 1]] || c[mid1] != c[mid2]) {
                                ++classes;
                        }
                        cn[p[i]] = classes - 1;
                }
                memcpy(c, cn, n * sizeof(int));
        }
        vector<int> minShifts;
        for (int i = 0; i < n; ++i) {
                if (minShifts.empty() || minShifts.back() > p[i]) {
                        minShifts.push_back(p[i]);
                }
        }
        int q;
        scanf("%d", &q);
        for (int i = 0; i < q; ++i) {
                int k;
                scanf("%d", &k);
                v[k].push_back(i);
        }
        int cur = INF;
        for (int k = 0; k < n; ++k) {
                while (!minShifts.empty() && minShifts.back() <= k) {
                        cur = minShifts.back();
                        minShifts.pop_back();
                }
                for (int i = 0; i < v[k].size(); ++i) {
                        ans[v[k][i]] = cur;
                }
        }
        for (int i = 0; i < q; ++i) {
                printf("%d\n", ans[i]);
        }
        return 0;
}
```

Tester Solution

```
#include <cstdio>
#include <iostream>
#include <algorithm>
#include <vector>
#include <queue>
#include <stack>
#include <set>
#include <map>
#include <cstring>
#include <cstdlib>
#include <cmath>
#include <string>
#include <memory.h>
#include <sstream>
#include <complex>

#define REP(i,n) for(int i = 0, _n = (n); i < _n; i++)
#define REPD(i,n) for(int i = (n) - 1; i >= 0; i--)
#define FOR(i,a,b) for (int i = (a), _b = (b); i <= _b; i++)
#define FORD(i,a,b) for (int i = (a), _b = (b); i >= _b; i--)
#define FORN(i,a,b) for(int i=a;i<b;i++)
#define FOREACH(it,c) for (__typeof((c).begin()) it=(c).begin();it!=(c).end();it++
#define RESET(c,x) memset (c, x, sizeof (c))

#define sqr(x) ((x) * (x))
#define PB push_back
#define MP make_pair
#define F first
#define S second
#define Aint(c) (c).begin(), (c).end()
#define SIZE(c) (c).size()

#define DEBUG(x) { cerr << #x << " = " << x << endl; }
#define PR(a,n) {cerr<<#a<<" = "; FOR(_,1,n) cerr << a[_] << ' '; cerr <<endl;}
#define PR0(a,n) {cerr<<#a<<" = ";REP(_,n) cerr << a[_] << ' '; cerr << endl;}
#define ll long long
using namespace std;

const double PI = 2.0 * acos (0.0);

typedef pair <int, int> PII;


#define SZ(x) ((int)(x.size()))

#define maxn 2000009
#define maxlogn 22

```

```
string s;
int n;
int sa[maxn],tam[maxn],inv[maxn],key[maxn],lcp[maxn],posa[maxn],myrank[maxn];
int rmq[maxn][maxlogn],LOG[maxn];
int ans[maxn];

void initSA() {
    int i,h,x;
    memset(tam,0,sizeof(tam));
    FOR(i,1,n) tam[s[i]]++;
    FOR(i,1,256) tam[i] += tam[i-1];
    cout << n << " done" << endl;
    FORD(i,n,1) sa[tam[s[i]]--] = i;
    x = 0;
    posa[0] = 1;
    key[sa[1]] = 0;
    FOR(i,2,n) {
        if (s[sa[i]] != s[sa[i-1]]) posa[++x] = i;
        key[sa[i]] = x;
    }
    h = 1;
    while (h < n) {

        FOR(i,1,n) tam[i]=sa[i];

        FOR(i,1,n) if (tam[i] > h) {
            x = tam[i] - h;
            sa[posa[key[x]]] = x;
            posa[key[x]]++;
        }

        x = 0;
        posa[0] = 1;
        tam[sa[1]] = 0;
        FOR(i,2,n) {
            if ((key[sa[i-1]] < key[sa[i]]) || ((key[sa[i-1]] == key[sa[i]]) &&
                posa[++x] = i;
            tam[sa[i]] = x;
        }
        FOR(i,1,n) key[i] = tam[i];
        if (x == n-1) break;
        h = h << 1;
    }

    FOR(i,1,n) myrank[sa[i]]=i;
}

void initLCP() {
    int i,j,h = 0,x;
    s[n + 1] = 0;
    int result = 0;
    FOR(i,1,n) inv[sa[i]] = i;
    FOR(i,1,n)
        if (inv[i] == 1) lcp[1] = 0;
        else {
            x = inv[i];
            j = sa[x - 1];
            while (s[j + h] == s[i + h]) h++;
            lcp[x] = h;
            if (h > 0) h--;
        }
}

void initRMQ() {
    LOG[1]=0;
    FOR(i,2,n) if((1<<(LOG[i-1]+1))==i) LOG[i]=LOG[i-1]+1; else LOG[i]=LOG[i-1];

    FOR(i,1,n) rmq[i][0]=lcp[i];

    FOR(j,1,LOG[n]) {
        FOR(i,1,n-(1<<j)+1) {
            rmq[i][j]=min(rmq[i][j-1],rmq[i+(1<<(j-1))][j-1]);
        }
    }
}

int getRMQ(int x,int y) {
    int len=LOG[y-x+1];
    return min(rmq[x][len],rmq[y-(1<<len)+1][len]);
}

int getLCP(int x,int y) {
    x=myrank[x];
    y=myrank[y];
    if(x==y) return n-x+1;
    if(x>y) swap(x,y);
    return getRMQ(x+1,y);
}

int main() {

    ios_base::sync_with_stdio(0);

    int N,q;

    cin >> N >> s >> q;
    s = " " + s;
    n=s.length();

    initSA();
    initLCP();
    initRMQ();

    int curind=1,curmin=inv[1];
    for(int i=2; i<=N; i++){
        if(inv[i]<curmin){
            if(getLCP(inv[i], curmin) < N)
                curind=i,curmin=inv[i];
        }
        ans[i]=curind;
    }
    for(int i=0; i<q; i++){
        int x;
        cin >> x;
        printf("%d\n",max(0,ans[x+1]-1));
    }

}
```

COMMENTS (2)   SORT BY: Relevance▾

Join Discussion...

Cancel   Post

SHUBHAM PANDEY 5 months ago

please explain something
this code dosen't mean anything

▲ 0 votes ● Reply ● Message ● Permalink

Prakhar Agrawal 4 months ago

How can we submit the editorial for the above question ?

▲ 0 votes ● Reply ● Message ● Permalink

ABOUT US

Blog

Engineering Blog

Updates & Releases

Team

Careers

In the Press

HACKEREARTH

API

Chrome Extension

CodeTable

HackerEarth Academy

Developer Profile

Resume

Get Badges

Campus Ambassadors

DEVELOPERS

AMA

Code Monk

Judge Environment

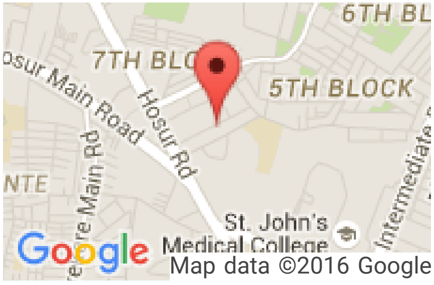
Solution Guide

Problem Setter Guide

Practice Problems

HackerEarth Challenges

College Challenges


6TH BL
7TH BLO
5TH BLOCK
Hosur Main Road
Hosur Rd
Main Rd
NTE
St. John's Medical College
Intermediate
Google
Map data ©2016 Google