4 LIVE EVENTS

Unique Gems

Attempted by: **170** / Accuracy: **76%** / ★★★★☆

Tag(s): Algorithms, Hard, Suffix Arrays

PROBLEM **EDITORIAL** MY SUBMISSIONS ANALYTICS

Setter's solution:

We need to count the number of substrings that occur exactly once.

This can be formulated as the number of substrings occuring >=1 times - number of substrings occuring >= 2 times.

The former is simply the number of distinct substrings.

The latter can be solved by observing the lcp array in detail.

IS THIS EDITORIAL HELPFUL?

Yes, it's helpful

No, it's not helpful

7 developer(s) found this editorial helpful.

Author Solution by Ankit Srivastava

```
t) {
```

```
| c == -1;
```

```
eamWriter(outputStream)));
```

```
At(o2));



 len characters
 first len characters





the end of the string
ollowed by 0-symbol
&& sa[i - 1] + len < n && r[sa[i - 1] + len / 2] == r[sa[i] + len / 2] ? rank[sa

s




characters
by second len characters

e already sorted
```

```
At(i + h) == s.charAt(j + h)) {
```

Tester Solution

```
#include <cstdio>
#include <cmath>
#include <iostream>
#include <set>
#include <algorithm>
#include <vector>
#include <map>
#include <cassert>
#include <string>
#include <cstring>
#include <queue>

using namespace std;

#define rep(i,a,b) for(int i = a; i < b; i++)
#define S2(x,y) scanf("%d%d",&x,&y)
#define P(x) printf("%d\n",x)
#define all(v) v.begin(),v.end()
#define sz size()
#define F first
#define S second

typedef long long int LL;
typedef pair<int, int > pii;
typedef vector<int > vi;

const int N = 100001;

```

```
string s;
vector<pii > ranks;

pair<int, pii > v[N];
int tag[N][22];
int mxPow;
int n;

void buildSA(string &s) {
        n = s.sz;
        rep(i,0,n) {
                v[i].F = s[i] - 'a' + 1;
                v[i].S.F = -1;
                v[i].S.S = i;
                tag[i][0] = v[i].F;
        }

        int len = 1;
        int j = 0;
        while(len < n) {
                len += len;
                j++;

                rep(i,0,n) {
                        v[i].F = tag[i][j-1];
                        if(i+len/2 < n) v[i].S.F = tag[i+len/2][j-1];
                        else v[i].S.F = -1;
                        v[i].S.S = i;
                }
                sort(v, v+n);

                int tagId = 1;
                tag[v[0].S.S][j] = 1;
                rep(i,1,n) {
                        if(v[i].F != v[i-1].F || v[i].S.F != v[i-1].S.F) {
                                tagId++;
                        }
                        tag[v[i].S.S][j] = tagId;
                }
        }
        mxPow = j;
}

int lcp(int x, int y) {

        int res = 0;
        for(int i = mxPow; i >= 0; i--) {
                if(tag[x][i] == tag[y][i]) {
                        res += 1 << i;
                        x += 1 << i;
                        y += 1 << i;
                }
                if(x >= n || y >= n) break;
        }
        return res;
}

int main() {
        int t;
        cin >> t;
        assert(t >= 1 && t <= 10);
        while(t--) {
                cin >> s;
                buildSA(s);
                int n = s.sz;
                assert(n > 0 && n <= 100000);

                ranks.clear();
                rep(i,0,n) {
                        ranks.push_back(make_pair(tag[i][mxPow], i));
                }
                // P(ranks.sz);
                sort(all(ranks));

                // cout << s << endl;
                // rep(i,0,n) printf("%d %d\n",ranks[i].F, ranks[i].S);
                // P(mxPow);

                LL ans = 0;
                rep(i,0,n) {
                        int mx = 0;
                        if(i-1 >= 0) mx = max(mx, lcp(ranks[i-1].S, ranks[i]
                        if(i+1 < n) mx = max(mx, lcp(ranks[i+1].S, ranks[i].
                        // P(mx);

                        ans += n - ranks[i].S - mx;
                }
                cout << ans << endl;
        }
        return 0;
}
```

Team
Careers
In the Press

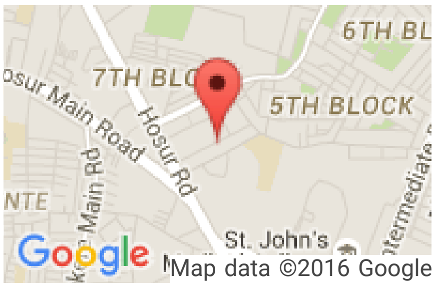
HackerEarth Academy
Developer Profile
Resume
Get Badges
Campus Ambassadors
Get Me Hired
Privacy
Terms of Service

Solution Guide
Problem Setter Guide
Practice Problems
HackerEarth Challenges
College Challenges
College Ranking
Organise Hackathon
Hackathon Handbook
Competitive Programming
Open Source

EMPLOYERS

Developer Sourcing
Lateral Hiring
Campus Hiring
Hackathons
FAQs
Customers

REACH US



IIIrd Floor, Salarpuria Business Center,
4th B Cross Road, 5th A Block,
Koramangala Industrial Layout,
Bangalore, Karnataka 560095, India.

contact@hackerearth.com
+91-80-4155-4695
+1-650-461-4192