2019 Multi-University Training Contest 2

比赛情况

题号	A	B	C	D	E	F	G	H	I	J	K	L
状态	-	O	-	-	O	-	-	O	-	O	O	O

O 在比赛中通过 Ø 赛后通过 ! 尝试了但是失败了 - 没有尝试

比赛时间

2020-07-09 13:00-18:00

提交记录

题解

E - Everything Is Generated In Equal Probability

给一个伪代码，然后给出一个 $N(N\le3000)$ 等概率的生成一个数 $n\in [1,N]$再等概率生成 n 的一个排列，让这个排列执行伪代码中的函数，问结果的期望是多少。

推，就硬推。考虑 n 一定时，存在 ${ n \choose 2}$ 个逆序对，每个逆序对出现的概率为 $\frac 12$按照上述函数每个逆序对对期望的贡献为:$\sum_{i=0}^\infty \frac 1{4^i}=\frac 43$所以总的期望为 ${n \choose 2}\frac12\frac43=\frac{n(n-1)}3$最后对每个 n 乘 $\frac1N$再相加即可。

```
#include<bits/stdc++.h>
#define ll long long
#define pii_ pair<int,int>
#define mp_ make_pair
#define pb push_back
#define fi first
#define se second
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define per(i,a,b) for(int i=(a);i>=(b);i--)
#define show1(a) cout<<#a<<" = "<<a<<endl
#define show2(a,b) cout<<#a<<" = "<<a<<"; "<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
const int maxn = 3e5+5;
const ll M = 998244353;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}
ll qpow(ll a,ll b) {ll
s=1;while(b){if(b&1)s=(s*a)%M;a=(a*a)%M;b>>=1;}return s; }
ll res[3005];
void init()
```

```
{
    int n = 3000;
    ll inv3 = qpow(3,M-2);
    rep(i,1,n){
        res[i] = i*(i-1)%M*inv3%M;
    }
}

int main()
{
    fastio();
    init(); int n;
    while(cin>>n){
        ll ans = 0;
        rep(i,1,n){
            ans = (ans + res[i])%M;
        }
        ans = ans * qpow(n,M-2) % M;
        cout<<ans<<endl;
    }
    return 0;
}
```

I - I Love Palindrome String

给一个字符串，对 $i\in [1,|S|]$ 统计有多少子串满足长度为 i 自身是一个回文串，该子串的前半段也是个回文串。

回文自动机求出所有回文串，字符串哈希看是否前半部分等于后半部分。

```
#include<bits/stdc++.h>
#define ll long long
#define ull unsigned long long
#define pii_ pair<int,int>
#define mp_ make_pair
#define pb push_back
#define fi first
#define se second
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define per(i,a,b) for(int i=(a);i>=(b);i--)
#define show1(a) cout<<#a<<" = "<<a<<endl
#define show2(a,b) cout<<#a<<" = "<<a<<"; "<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
const int maxn = 3e5+5;
```

```
const ull base = 19260817;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}
ull hs[maxn],bs[maxn];
char s[maxn];ll ans[maxn];
inline ull HASH(int l,int r) {return hs[r] - hs[l-1]*bs[r-l+1];}
struct Palindromes_Automation
{
    int
last,n,sz,trans[maxn][26],fail[maxn],cnt[maxn],len[maxn],s[maxn],pos[maxn];
    char t[maxn];
    void init()
    {
        memset(trans,0,sizeof(trans));
        memset(cnt,0,sizeof(cnt));
        memset(len,0,sizeof(len));
        memset(ans,0,sizeof(ans));
        last=0,sz=1;
        len[0] = 0,len[1] = -1;
        fail[0] = 1,fail[1] = 0;
    }
    int get_fail(int x) {while(s[n]!=s[n-len[x]-1])x=fail[x];return x;}
    void extend()
    {
        int fa = get_fail(last);
        int c = s[n];
        if(!trans[fa][c]){
            len[++sz] = len[fa] + 2;
            fail[sz] = trans[get_fail(fail[fa])][c];
            trans[fa][c] = sz;
        }
        last = trans[fa][c];
        cnt[last]++;
        pos[last] = n;
    }
    void solve(char str[])
    {
        memcpy(t,str,sizeof(t));
        int l = strlen(t+1); s[0] = 26;
        rep(i,1,l) hs[i] = hs[i-1]*base + t[i];
        for(n=1;n<=l;n++){
            s[n] = t[n] - 'a';
            extend();
        }
        per(i,sz,0) cnt[fail[i]] += cnt[i];
        rep(i,2,sz){
            int tmp = pos[i];
            if(len[i]%2==0){
                if(HASH(tmp-len[i]+1,tmp-len[i]/2)==HASH(tmp-
len[i]/2+1,tmp)) ans[len[i]]+=cnt[i];
            }else{
                if(HASH(tmp-len[i]+1,tmp-len[i]/2)==HASH(tmp-len[i]/2,tmp))
```

```
ans[len[i]]+=cnt[i];
            }
        }
        rep(i,1,l){
            cout<<ans[i];
            if(i==l) cout<<endl;
            else cout<<" ";
        }
    }
}pam;

int main()
{
    fastio();
    bs[0] = 1; rep(i,1,maxn-1) bs[i] = bs[i-1]*base;
    while(cin>>s+1){
        pam.init();
        pam.solve(s);
    }
    return 0;
}
```

replay

比赛总结