

知识点 - 回文树

解决问题类型：

所有回文串问题

定义与代码：

```
const int maxn = 1e5, sigma = 26;

int s[maxn], len[maxn], link[maxn], to[maxn][sigma];

int n, last, sz;

void init()
{
    s[n++] = -1;
    link[0] = 1;
    len[1] = -1;
    sz = 2;
}

int get_link(int v)
{
    while(s[n - len[v] - 2] != s[n - 1]) v = link[v];
    return v;
}

void add_letter(int c)
{
    s[n++] = c;
    last = get_link(last);
    if(!to[last][c])
    {
        len[sz] = len[last] + 2;
        link[sz] = to[get_link(link[last])][c];
        to[last][c] = sz++;
    }
    last = to[last][c];
}
```

例题

[CF17E Palisection](#)

题意

求字符串中相交（含公共部分，所以包含也算相交）的回文子串的对数。模一个并不是质数的数。

题解

正难则反吧。总回文子串的个数就是回文树上的 $\sum \text{depi}$ 所以总对数就是 $1/2 \sum \text{depi} * (\sum \text{depi} - 1)$ 然后就是要求不相交的回文串对数。枚举前面那个回文子串的终止位置 i ，那么后一个回文子串的起始位置至少在 $i+1$ 之后。所以正反建两遍回文树，求出以每一个 i 为起始位置/终止位置的回文串个数，然后乘一下就行了。

```
#include<cstdio>
#include<algorithm>
#include<cstring>
using namespace std;
const int N = 2e6+5;
const int mod = 51123987;
int n, fa[N], len[N], dep[N], tot, last, p1[N], p2[N], ans;
int to[N], nxt[N], ww[N], head[N], cnt;
char s[N];
void init()
{
    fa[last=0]=fa[1]=1;
    len[tot=1]=-1;
    memset(head, 0, sizeof(head)); cnt=0;
}
void link(int u, int v, int c)
{
    to[++cnt]=v; nxt[cnt]=head[u]; ww[cnt]=c;
    head[u]=cnt;
}
int tr(int v, int c)
{
    for (int e=head[v]; e; e=nxt[e])
        if (ww[e]==c) return to[e];
    return 0;
}
void extend(int c, int n)
{
    int v=last;
    while (s[n-len[v]-1]!=s[n]) v=fa[v];
    if (!tr(v, c))
    {
        int u=++tot, k=fa[v];
        len[u]=len[v]+2;
        while (s[n-len[k]-1]==s[n]) k=fa[k];
        fa[u]=tr(k, c); dep[u]=dep[fa[u]]+1;
        link(v, u, c);
    }
    last=tr(v, c);
}
int main()
{
    scanf("%d", &n);
    scanf("%s", s+1);
    init();
    for (int i=1; i<=n; ++i) extend(s[i]-'a', i), (ans+=(p1[i]=dep[last]))%=mod;
    ans=1ll*ans*(ans-1)/2%mod;
    reverse(s+1, s+n+1);
    for (int i=1; i<=n; ++i) extend(s[i]-'a', i), (ans+=(p2[i]=dep[last]))%=mod;
    ans=1ll*ans*(ans-1)/2%mod;
    printf("%d", ans);
}
```

```
init();  
for (int i=1;i<=n;++i) extend(s[i]-'a',i),p2[n-i+1]=dep[last];  
for (int i=n;i--i) (p2[i]+=p2[i+1])%=mod;  
for (int i=1;i<=n;++i) ans=(ans-1ll*p1[i]*p2[i+1]%mod+mod)%mod;  
printf("%d\n",ans);  
return 0;  
}
```