

```

1  const int N=2e6+5;
2  int ch[N][26],l[N],fa[N],cnt,root,last;
3  char s[N];
4  void init(){last=root=++cnt;}
5  void insert(int x){//字符的位置
6      int p=last,np=++cnt,c=s[x]-'a';l[np]=x;last=np;
7      for (;p && !ch[p][c];p=fa[p])ch[p][c]=np;
8      if (!p)fa[np]=root;
9      else {
10         int q=ch[p][c];
11         if (l[p]+1==l[q])fa[np]=q;
12         else {
13             int nq=++cnt;l[nq]=l[p]+1;
14             memcpy(ch[nq],ch[q],sizeof(ch[q]));
15             fa[nq]=fa[q];fa[np]=fa[q]=nq;
16             for (;ch[p][c]==q;p=fa[p])ch[p][c]=nq;
17         }
18     }
19 }

```

```

1  const long long mod=998244353;
2  long long pw(long long x,long long y)
3  {
4      y=(y%(mod-1)+mod-1)%(mod-1);
5      long long res=1;
6      while(y>0)
7      {
8          if(y&1) res=res*x%mod;
9          x=x*x%mod;
10         y>>=1;
11     }
12     return res;
13 }
14 void NTT(long long *A,int len,int opr)
15 {
16     for(int i=1,t=0;i<len;i++)
17     {
18         int k=len;
19         do t^=(k>=1); while(~t&k);
20         if(i<t) swap(A[i],A[t]);
21     }
22     for(int h=2;h<=len;h<=1)
23     {
24         long long wn=pw(3,(mod-1)/h*opr);
25         for(int i=0;i<len;i+=h)
26         {
27             long long w=1;

```

```

28         for(int j=i;j<i+(h>>1);j++)
29         {
30             long long t1=A[j],t2=A[j+
(h>>1)]*w%mod;
31             A[j]=(t1+t2)%mod;
32             A[j+(h>>1)]=(t1-t2+mod)%mod;
33             w=w*wn%mod;
34         }
35     }
36 }
37 if(opr<0)
38     for(int i=0;i<len;i++)
39         A[i]=(A[i]*((1-mod)/len)%mod+mod)%mod;
40 }

```

```

1  typedef complex<double> cd;
2  typedef vector<cd> vcd;
3  void fft(vcd &A,int k){
4      int As=A.size();
5      if(As==1) return ;
6      vcd B0(As/2),B1(As/2);
7      for(int i=0;i<As;i++) if(i&1) B1[i/2]=A[i];else
B0[i/2]=A[i];
8      fft(B0,k);fft(B1,k);
9      cd
i(0,1),wn(exp((k*2*acos(-1)/(double)As)*i)),w(1,0);
10     for(int i=0;i<As/2;i++){
11         A[i]=B0[i]+w*B1[i];
12         A[i+As/2]=B0[i]-w*B1[i];
13         w=w*wn;
14     }
15 }
16
17

```

```

//baby_step giant_step
//  $a^x = b \pmod n$  n是素数和不是素数都可以
// 求解上式  $0 \leq x < n$  的解
#define MOD 76543
int hs[MOD], head[MOD], next[MOD], id[MOD], top;
void insert(int x, int y)
{
    int k = x % MOD;
    hs[top] = x, id[top] = y, next[top] = head[k], head[k] = top++;
}
int find(int x)
{
    int k = x % MOD;
    for(int i = head[k]; i != -1; i = next[i])
        if(hs[i] == x)
            return id[i];
    return -1;
}
int BSGS(int a, int b, int n)
{
    memset(head, -1, sizeof(head));
    top = 1;
    if(b == 1) return 0;
    int m = sqrt(n * 1.0), j;
    long long x = 1, p = 1;
    for(int i = 0; i < m; ++i, p = p * a % n) insert(p * b % n, i);
    for(long long i = m; ; i += m)
    {
        if( (j = find(x = x * p % n)) != -1 ) return i - j;
        if(i > n) break;
    }
    return -1;
}

```