

回文自动机

回文自动机

luoguP5496 【模板】回文自动机 (PAM)

2022牛客多校9 G-Magic Spells

luoguP1659 [国家集训队]拉拉队排练

P4287 [SHOI2011]双倍回文

luoguP5496 【模板】回文自动机 (PAM)

给定一个字符串 s ,求以 s 的每个位置结尾的回文子串个数。

```
#include <cstdio>
#include <cstring>
const int N=5e5+7,M=30;
char str[N];
struct Palindrome_Automaton{
    int Size,Last,Root0,Root1,Trans[N][M],Link[N];
    int Len[N],Cnt[N];
    Palindrome_Automaton(){
        Root0=Size++,Root1=Size++;
        Last=Root1;
        Len[Root0]=0,Link[Root0]=Root1;
        Len[Root1]=-1,Link[Root1]=Root1;
    }
    void Extend(int ch,int dex){
        int u=Last;
        while(str[dex-Len[u]-1]!=str[dex]) u=Link[u]; //getfail
        if(!Trans[u][ch]){
            int Newnode=++Size,v=Link[u];
            Len[Newnode]=Len[u]+2;
            while(str[dex-Len[v]-1]!=str[dex]) v=Link[v];
            Link[Newnode]=Trans[v][ch],Trans[u][ch]=Newnode;
            Cnt[Newnode]=Cnt[Link[Newnode]]+1;
        }
        Last=Trans[u][ch];
    }
    void Build(char *str){
        int len=strlen(str);
        for(int i=0;i<len;i++){
            Extend(str[i]-'a'+1,i);
            str[i+1]=(str[i+1]-97+Cnt[Last])%26+97; //这是模板题的要求
            printf("%d ",Cnt[Last]);
        }
    }
}PAM;

signed main(){
    scanf("%s",str);
    PAM.Build(str);
    return 0;
}
```

```
}
```

2022牛客多校9 G-Magic Spells

给定 k 个串 $\{S_i\}$,问每个串中都出现的本质不同回文子串个数。 $\sum |S_i| \leq 3 \times 10^5, k \leq 5$

对每个串都建立一个PAM, 然后同时对 k 个PAM进行遍历, 只向每个PAM都能扩展出来的字符出边走。

```
#include<bits/stdc++.h>
using namespace std;
const int N=6e5+7;

int read(){ int x=0,f=1;char ch=getchar();while(ch<'0' || ch>'9'){if(ch=='-')
f=f*-1;ch=getchar();}while(ch>='0'&&ch<='9'){x=x*10+ch-'0';ch=getchar();}return
x*f;}
void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}

int n,m,f[N][10],ans=0;
char s[N];
int ch[N][26],fa[N],len[N];

struct E{
    int v,nxt;
}e[N<<1];
int ecnt=0,head[N];
inline void addedge(int u,int v){e[++ecnt]=(E){v,head[u]};head[u]=ecnt;}
int lst,cnt=1;

inline void init(){
    s[0]=-1;
    len[1]=-1;
    fa[0]=fa[1]=1;
    lst=0;
}
inline void insert(int k,int c,int id){
    int p=lst;
    while(s[k-len[p]-1]!=c+'a')p=fa[p];
    if(!ch[p][c]){
        int q=lst==++cnt?v=fa[p];
        while(s[k-len[v]-1]!=c+'a') v=fa[v];
        fa[q]=ch[v][c];
        len[ch[p][c]=q]=len[p]+2;
    } else lst=ch[p][c];
    f[lst][id]++;
}
inline void dfs(int x){
    for(int i=head[x];i;i=e[i].nxt){
        int y=e[i].v;
        dfs(y);
        for(int k=0;k<n;++k){
            f[x][k]+=f[y][k];
        }
    }
}
int flag=0;
```

```

    for(int k=0;k<n;++k) if(!f[x][k]) flag=1;
    if(x==1||x==0||flag) return;
    ans++;
}

signed main(){
    scanf("%d",&n);
    for(int i=0;i<n;++i){
        init();
        scanf("%s",s);
        for(int j=0;j<strlen(s;++j) insert(j,s[j]-'a',i);
    }
    addedge(1,0);
    for(int i=2;i<=cnt;++i) addedge(fa[i],i);
    dfs(1);
    write(ans),putchar('\n');
    return 0;
}

```

luoguP1659 [国家集训队]拉拉队排练

求所有奇长回文串，并将长度连乘作为答案。

可以用马拉车搞，这里贴一手PAM的代码。(需要卡空间)

```

#include<algorithm>
#include<stdio.h>
#include<cstring>
typedef long long ll;
const int N=1e6+1,M=26;
const int mod=19930726;

ll fpow(ll a,int b){
    ll res=1;
    while(b){
        if(b&1) res=res*a%mod;
        a=a*a%mod;
        b>>=1;
    }
    return res;
}

int n,k;
namespace PAM{
    int Size,Last;
    char str[N];
    struct nod{
        int ch[M],fa,len,sz;
        nod(){len=fa=0;memset(ch,0,sizeof(ch));}
    }prt[N];
    bool cmp(nod A,nod B){return A.len>B.len;}
    void Init(){ //初始化PAM
        Last=Size=1;
        prt[0].len=0;prt[0].fa=1;
        prt[1].len=-1;prt[1].fa=1;
    }
}

```

```

int Getfail(int x){
    while(str[n-prt[x].len-1]!=str[n]) x=prt[x].fa;
    return x;
}
void Extend(int x){
    int cur=Getfail(Last);
    if(!prt[cur].ch[x]){
        int now=++Size;
        prt[now].len=prt[cur].len+2;
        prt[now].fa=prt[Getfail(prt[cur].fa)].ch[x];
        prt[cur].ch[x]=now;
    }
    prt[prt[cur].ch[x]].sz++;
    Last=prt[cur].ch[x];
}
};
using namespace PAM;

signed main(){
    int Leng=0;
    scanf("%d%d",&Leng,&k);
    Init();
    scanf("%s",str);
    for(n=0;n<Leng;n++){Extend(str[n]-'a');}
    for(int i=Size;i>=2;--i) prt[prt[i].fa].sz=
(prt[prt[i].fa].sz+prt[i].sz)%mod;
    std::stable_sort(prt+1,prt+Size+1,cmp);
    int now=1;ll ans=1;
    while(k){ //统计答案
        if(now>Size){
            printf("-1\n");
            return 0;
        }
        if(prt[now].len%2==0){
            ++now;
            continue;
        }
        if(prt[now].sz<k){
            k-=prt[now].sz;
            ans*=fpow(prt[now].len,prt[now].sz)%mod;
            ans%=mod;
            ++now;
        }else{
            ans*=fpow(prt[now].len,k)%mod;
            ans%=mod;
            k=0;
        }
    }
    printf("%lld\n",ans);
    return 0;
}

```

P4287 [SHOI2011]双倍回文

给定长度为 n 的字符串 s ，求双倍回文子串，即 ww^Rww^R 形式的回文串。

```
#include<algorithm>
#include<stdio.h>
#include<cstring>
typedef long long ll;
const int N=5e5+10,M=26;

int n,f[N];
namespace PAM{
    int Size,Last;
    char str[N];
    struct nod{
        int ch[M],fa,len,sz;
        nod(){len=fa=0;memset(ch,0,sizeof(ch));}
    }prt[N];
    bool cmp(nod A,nod B){return A.len>B.len;}
    void Init(){ //初始化PAM
        Last=0;Size=1;
        prt[0].len=0;prt[0].fa=1;
        prt[1].len=-1;prt[1].fa=1;
    }
    int Getfail(int x){
        while(str[n-prt[x].len-1]!=str[n]) x=prt[x].fa;
        return x;
    }
    void Extend(int x){
        int cur=Getfail(Last);
        if(!prt[cur].ch[x]){
            int now=++Size;
            prt[now].len=prt[cur].len+2;
            prt[now].fa=prt[Getfail(prt[cur].fa)].ch[x];
            prt[cur].ch[x]=now;
            if(prt[prt[now].fa].len<=prt[now].len>>1) f[now]=prt[now].fa; //注意
            这里,给满足长度关系的结点连边!!!
            else{
                int p=f[cur];
                while((prt[p].len+2>prt[now].len>>1)|| (str[n]!=str[n-prt[p].len-1])) p=prt[p].fa;
                f[now]=prt[p].ch[x];
            }
        }
        prt[prt[cur].ch[x]].sz++;
        Last=prt[cur].ch[x];
    }
};
using namespace PAM;

signed main(){
    int Leng=0,ans=0;
    scanf("%d",&Leng);
    Init();
    scanf("%s",str);
    for(n=0;n<Leng;n++) Extend(str[n]-'a');
    for(int i=2;i<=Size;++i){
```

```
        if(prt[i].len%4==0&&(prt[f[i]].len==prt[i].len>1))
ans=std::max(ans,prt[i].len);
    }
    printf("%d\n",ans);
    return 0;
}
```