# Huiwen

3676

#include<bits/stdc++.h>

long long ans; int n;

char ch[300005];

struct pam{

int cnt,last;

int a[300005][26],fa[300005],l[300005],size[300005];

pam(){

cnt=1;fa[0]=fa[1]=1;l[1]=-1;

}

void extend(int c,int n){

int p=last;

while(ch[n-l[p]-1]!=ch[n])p=fa[p];

if(!a[p][c])

{

int now=++cnt,k=fa[p];

l[now]=l[p]+2;

while(ch[n-l[k]-1]!=ch[n])k=fa[k];

fa[now]=a[k][c];a[p][c]=now;

}

last=a[p][c];

size[last]++;

}

void solve(){

for(int i=cnt;i;i--){

size[fa[i]]+=size[i];

ans=max(ans,(ll)size[i]\*l[i]);

}

}

}pam;

int main(){

scanf("%s",ch+1);

n=strlen(ch+1);

for(int i=1;i<=n;i++)

pam.extend(ch[i]-'a',i);

pam.solve();

printf("%lld\n",ans);

return 0;

}

3790

#include<iostream>

#include<cstring>

#include<cstdio>

#include<algorithm>

#include<cmath>

#define ll long long

#define inf 1000000000

using namespace std;

inline int read()

{

int x=0;char ch=getchar();

while(ch<'0'||ch>'9')ch=getchar();

while(ch>='0'&&ch<='9'){x=x\*10+ch-'0';ch=getchar();}

return x;

}

int n,m,cnt;

int p[200005],t[100005];

char ch[100005],a[200005];

struct seg{

int l,r;

}l[100005];

int query(int x)

{

if(!x)return 0;

int tmp=inf;

for(int i=x;i<=n;i+=i&-i)tmp=min(t[i],tmp);

return tmp;

}

void modify(int x,int val)

{

for(int i=x;i;i-=i&-i)t[i]=min(t[i],val);

}

bool operator<(seg a,seg b)

{

return a.r<b.r;

}

void add(int x,int y)

{

x=x/2+1;y=y/2-1;

if(x>y)return;

l[++cnt]=(seg){x,y};

}

void manacher()

{

m=2\*n+1;

for(int i=1;i<=n;i++)

{

a[i<<1]=ch[i];

a[i<<1|1]='#';

}

a[0]='+';a[1]='#';a[m+1]='-';

int mx=0,id;

for(int i=1;i<=m;i++)

{

if(mx>i)p[i]=min(mx-i,p[2\*id-i]);

else p[i]=1;

for(;a[i-p[i]]==a[i+p[i]];p[i]++);

add(i-p[i],i+p[i]);

if(p[i]+i>mx)mx=p[i]+i,id=i;

}

}

int main()

{

while(scanf("%s",ch+1)!=EOF)

{

cnt=0;

n=strlen(ch+1);for(int i=1;i<=n;i++)t[i]=inf;

manacher();

sort(l+1,l+cnt+1);

int ans=inf;

for(int i=1;i<=cnt;i++)

{

int x=query(l[i].l-1)+1;

modify(l[i].r,x);

if(l[i].r==n)ans=min(ans,x);

}

printf("%d\n",ans-1);

}

return 0;

}