**SAM**

const int MAXL=2000005;

const int MAXCH=26;

char s[MAXL];

char CH='a';

int tot=0;//tot为总结点数

int maxlen[MAXL],minlen[MAXL],trans[MAXL][MAXCH],slink[MAXL];

**/\***

**一些非必要的功能属性**

**若去掉 还需在函数中修改**

**\*/**

int ind[MAXL];//树中子节点个数

int edpts[MAXL];//结点的endpos个数

bool pre[MAXL];//是否为前缀所在结点

**/\***

**声明新结点的函数 未知初值的 值可用-1代替 未知初值的引用 可用NULL代替**

**\*/**

int new\_state(int \_maxlen,int \_minlen,int \*\_trans,int \_slink)

{

maxlen[tot]=\_maxlen;

minlen[tot]=\_minlen;

for(int i=0;i<MAXCH;i++)

{

if(\_trans==NULL)

trans[tot][i]=-1;

else trans[tot][i]=\_trans[i];

}

slink[tot]=\_slink;

return tot++;

}

**/\***

**在u结点后加入字符ch**

**\*/**

int add\_char(char ch,int u)

{

int c=ch-CH;

int z=new\_state(maxlen[u]+1,-1,NULL,-1);

pre[z]=true;//该结点必为包含原串前缀的结点

int v=u;

while(v!=-1&&trans[v][c]==-1)

{

trans[v][c]=z;

v=slink[v];

}

if(v==-1)//最简单的情况，suffix-path(u->s)上都没有对应字符ch的转移

{

minlen[z]=1;

slink[z]=0;

++ind[0];

return z;

}

int x=trans[v][c];

if(maxlen[v]+1==maxlen[x])//较简单的情况，不用拆分x

{

minlen[z]=maxlen[x]+1;

slink[z]=x;

++ind[x];

return z;

}

int y=new\_state(maxlen[v]+1,-1,trans[x],slink[x]);//最复杂的情况，拆分x

slink[y]=slink[x];

ind[y]+=2;

minlen[x]=maxlen[y]+1;

slink[x]=y;

minlen[z]=maxlen[y]+1;

slink[z]=y;

int w=v;

while(w!=-1&&trans[w][c]==x)//该部分字符串长度越来越短 一定都是要转移到y的 因为y保留的是x原本较短的部分

{

trans[w][c]=y;

w=slink[w];

}

minlen[y]=maxlen[slink[y]]+1;

return z;

}

**/\***

**拓扑排序**

**获得每个结点的endpos个数**

**该种topsort是从叶子结点往上的排序 下文有从根节点往下的例子**

**\*/**

void getEndPtCount() {

queue<int> q;

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

if(!ind[i])

q.push(i);

while( !q.empty() ) {

int u = q.front();

q.pop();

if(pre[u])edpts[u]++;

edpts[ slink[u]] += edpts[u];

if( !--ind[slink[u]] ) q.push(slink[u]);

}

}

/\*

使用示例：

求解字符串中所有不同子串个数 hihocoder 1445

& 求解字符串中每个长度子串出现最多的次数 hihocoder 1449

\*/

int anlen[MAXL];

int n;

ll an;

int main()

{

/\*

初始化部分

\*/

int pres=new\_state(0,0,NULL,-1);

/\*

通用过程代码部分

\*/

scanf("%s",s);

// tot=1;//tot=0的表示空串的结点 故需先设tot=1 使得新结点从1开始编号

n=strlen(s);

for(int i=0;i<n;i++)

pres=add\_char(s[i],pres);//逐个加入结点

/\*

求解字符串中所有不同子串个数

\*/

for(int i=1;i<tot;i++)//所有结点下标范围为 [0,tot) 0为空串结点 故从1开始计算

an+=1LL\*maxlen[i]-minlen[i]+1;

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

/\*

求解字符串中每个长度子串出现最多的次数

\*/

getEndPtCount();

for(int i=1;i<tot;i++)//所有结点下标范围为 [0,tot) 0为空串结点 故从1开始计算

anlen[maxlen[i]]=max(anlen[maxlen[i]],edpts[i]);

for(int i=n;i>=1;i--)

anlen[i]=max(anlen[i],anlen[i+1]);

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

printf("%d\n",anlen[i]);

}