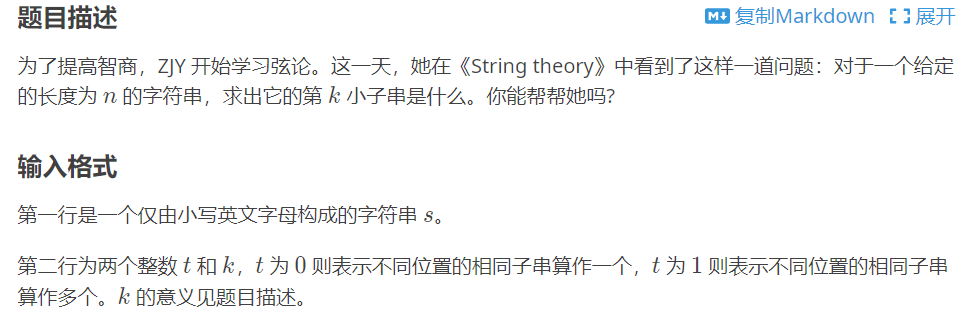
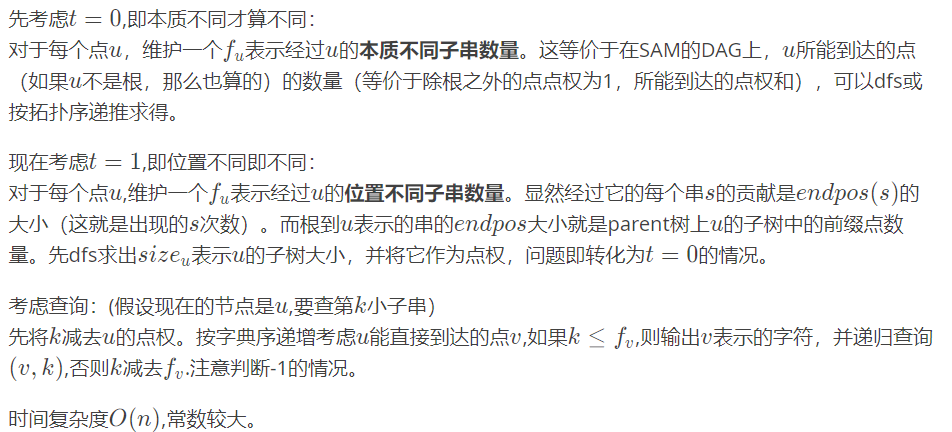
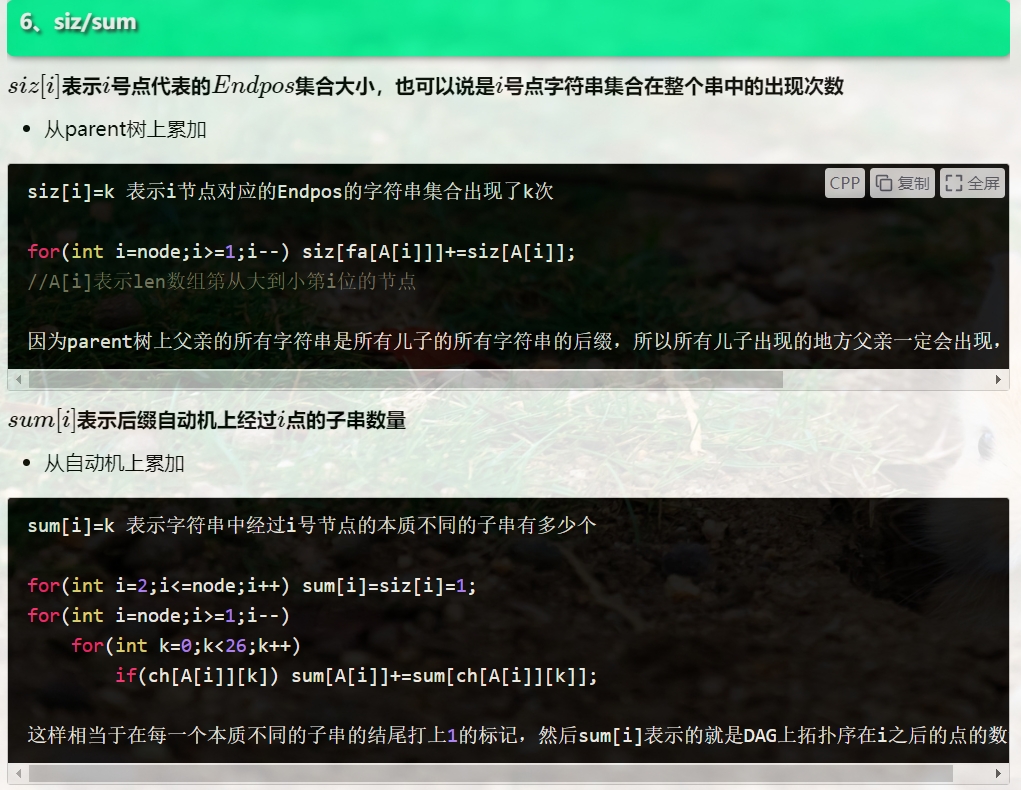
P3975 [TJOI2015]弦论



题解：



这个题解中的f[]即为下面代码中的sum[]



#include<bits/stdc++.h>

#define ll long long

using namespace std;

const int N=1100000;

char s[N];

int fa[N],len[N],siz[N],ch[N][26];

int t[N],A[N];

ll sum[N],K;

int l,lst=1,node=1,T;

void Extend(int c)//插入字符

{

int f=lst,p=++node;lst=p;

len[p]=len[f]+1;siz[p]=1;

while(f&&!ch[f][c]) ch[f][c]=p,f=fa[f];

if(!f) {fa[p]=1;return;}

int x=ch[f][c],y=++node;

if(len[x]==len[f]+1) {fa[p]=x;node--;return;}

memcpy(ch[y],ch[x],sizeof(ch[y]));

len[y]=len[f]+1; fa[y]=fa[x]; fa[x]=fa[p]=y;

while(f&&ch[f][c]==x) {ch[f][c]=y;f=fa[f];}

}

//前面都差不多

void Print(int x,int k)//输出字符串，这个过程和trie树上的查找某一个子串很相似

{

if(k<=siz[x]) return;

k-=siz[x];

for(int i=0;i<26;i++)

{

int R=ch[x][i];if(!R) continue;

if(k>sum[R]){k-=sum[R];continue;}

putchar(i+'a');Print(R,k);return;

}

}

int main()

{

//Part 1 Build SAM

scanf("%s%d%lld",s,&T,&K);l=strlen(s);

for(int i=l;i>=1;i--)s[i]=s[i-1];s[0]=0;

for(int i=1;i<=l;i++)Extend(s[i]-'a');

//Part 2 Sort

for(int i=1;i<=node;i++)t[len[i]]++;

for(int i=1;i<=node;i++)t[i]+=t[i-1];

for(int i=1;i<=node;i++)A[t[len[i]]--]=i;

//基数排序，变成逆拓扑序

for(int i=node;i>=1;i--)siz[fa[A[i]]]+=siz[A[i]];

for(int i=1;i<=node;i++)T==0?(sum[i]=siz[i]=1):(sum[i]=siz[i]);//根据T来调整sum[]

siz[1]=sum[1]=0;

/\*

这一段代码调试了半个小时

前者是对自动机处理（自动机上累加求的是子串个数）

后者是parent树（parent树上累加求的是i节点对应的endpos的字符集的longest的出现次数）

\*/

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

for(int j=0;j<26;j++)

if(ch[A[i]][j])sum[A[i]]+=sum[ch[A[i]][j]];

//最后还是要通过sum在自动机(可以理解为一个被压缩的trie树)上的dp来得出答案，

if(sum[1]<K)puts("-1");

else Print(1,K),puts("");

return 0;

}