序列自动机(Se AM)的总结

听起来很高大上，实际上代码简单的不行。

刚刚学，整理一下原理和应用。

原理：用一个数组$next[i][j]$维护第i位 (从下标1开始) 字符后 （也就是从第$i+1$位开始)

离第i位最近的字符为$ch=' a'+j$的位置。

可能数组定义有点复杂，举个例子就明白了。

$string:abcda$

对于第一个字符a:

$nt[1][1]=5$，即离第一个字符a最近的a的位置是5。

$nt[1][2]=2$,即离第一个字符a最近的b的位置是2。

依次类推：$nt[1][3]=3,nt[1][4]=4,nt[1][5]=0(\text{不存在})\dots$

这样维护一个字符串有什么好处呢，显然对于寻找某个子序列或者子串的时候非常方便。

应用1.判断字符串是否为某一个字符串(文本串)的子序列或者子串。

ep:我们要给的 $s=abcda$, 待查询的字符串 $t=da$。

初始位置$p=0$。

对于$t_1=d,next[p][d-' a']=4\rightarrow p=4\rightarrow next[p][a-' a']=5$.完毕

十分方便。

时间复杂度：$O(n+m)$,n是文本串长度，m是所以待查询串总长度。

求$next[][]$的两种写法。

```
void SeAM(){
    int len=strlen(a+1);
    for(int i=len;i;i--){
        for(int j=0;j<26;j++) nt[i-1][j]=nt[i][j];
        nt[i-1][a[i]-'a']=i;
    }
}
////////////////////
void SeAM(){
    int len=strlen(a+1);
    for(int i=len;i;nt[i-1][a[i]-'a']=i,i--)
        memcpy(nt[i-1],nt[i],sizeof nt[i]);
}
```

例题1：判断是否为子序列

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e6+5,M=1e6+5,inf=0x3f3f3f3f,mod=1e9+7;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
char a[N],b[N];
int nt[N][26],q;
void SeAM(){
    int len=strlen(a+1);
    for(int i=len;i;i--){
        for(int j=0;j<26;j++) nt[i-1][j]=nt[i][j];
        nt[i-1][a[i]-'a']=i;
    }
}
int main(){
    scanf("%s%d",a+1,&q);
    SeAM();
    while(q--){
        scanf("%s",b+1);
        int len=strlen(b+1),p=0,f=0;
        for(int i=1;i<=len;i++){
             p=nt[p][b[i]-'a'];
             if(!p) {
                 f=1;break;
             }
        }
        puts(f?"No":"Yes");
    }
    return 0;
}
```

例题2：判断是否为子串
和上面代码其实是一样的。

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e6+5,M=1e6+5,inf=0x3f3f3f3f,mod=1e9+7;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
char a[N],b[N];
int nt[N][26],tmp[26],q;
void SeAM(){
    int len=strlen(a+1);
    for(int i=len;i;nt[i-1][a[i]-'a']=i,i--)
```

```
        memcpy(nt[i-1],nt[i],sizeof nt[i]);
}
int main(){
    scanf("%s%d",a+1,&q);
    SeAM();
    while(q--){
        scanf("%s",b+1);
        int len=strlen(b+1),p=0,f=0;
        for(int i=1;i<=len;i++){
             p=nt[p][b[i]-'a'];
             if(!p) {
                 f=1;break;
             }
        }
        puts(f?"No":"Yes");
    }
    return 0;
}
```

例题3：判断子序列并记数

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=5e4+5,M=1e6+5,inf=0x3f3f3f3f,mod=1e9+7;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
int nt[N][26],n;
char a[N],b[N];
void SeAM(){
    int len=strlen(a+1);
    for(int i=len;i;nt[i-1][a[i]-'a']=i,i--)
        memcpy(nt[i-1],nt[i],sizeof nt[i]);
}
int main(){
    scanf("%s%d",a+1,&n);
    SeAM();
    int ans=0;
    for(int i=1;i<=n;i++){
        scanf("%s",b+1);
        int len=strlen(b+1),p=0,ok=1;
        for(int i=1;i<=len;i++){
            p=nt[p][b[i]-'a'];
            if(!p){
                ok=0;
                break;
            }
        }
        if(ok) ans++;
    }
    printf("%d\n",ans);
```

```
    return 0;
}
```

应用2：求k个字符串的公共子序列个数。

$ep:k=2,string:a,b..$

根据定义我们可以设$f[x][y]$表示a从第x位开始和b从第y位的公共子序列个数。

显然我们可以枚举公共子序列分别为$a,b,c\ldots,z$的贡献，然后求和。

根据$next[][]$数组的定义，我们可以的递推式。

如果$next_a[x][ch]\&\&next_b[y][ch],令nx=next_a[x][ch],ny=next_b[y][ch].$

有状态转移方程：$f[x][y]+=f[nx][ny]$

然后不断递归就行了，初始状态是如果x,y都不为0，因为他们的首字母相同，一个字母也构成了公共子序列，即$f[x][y]++$。

$ep:string_a=abcd,string_b=bc$。

显然公共子序列是：

(b,b)
(c,c)
(bc,bc)

我们模拟一下递归过程：
$dfs(0,0)\to dfs(2,1)\to dfs(3,2)$ 结束。

递归到最深处开始回溯：$f[3][2]=1\to f[2][1]=2\to f[0][0]=3$结束。

时间复杂度：$O(n^3+3n)$
如果还不理解，可以自己在多打印几次递归。

例题：3个字符串的公共子序列个数

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=150+5,M=1e6+5,inf=0x3f3f3f3f,mod=1e8;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
char a[N],b[N],c[N];
int nta[N][26],ntb[N][26],ntc[N][26],q,f[N][N][N],n;
void SeAM(char *a,int nt[][26]){
    int len=strlen(a+1);
    for(int i=len;i;nt[i-1][a[i]-'a']=i,i--)
        memcpy(nt[i-1],nt[i],sizeof nt[i]);
}
int dfs(int x,int y,int z){
    if(f[x][y][z]) return f[x][y][z];
    //printf("(%d,%d)\n",x,y);
    for(int i=0;i<26;i++){
```

```
        if(nta[x][i]&&ntb[y][i]&&ntc[z][i])
        f[x][y][z]=(f[x][y][z]+dfs(nta[x][i],ntb[y][i],ntc[z][i]))%mod;
    }
    if(x&&y&&z) ++f[x][y][z];
    return f[x][y][z]%mod;
}
int main(){
    scanf("%d%s%s%s",&n,a+1,b+1,c+1);
    SeAM(a,nta);
    SeAM(b,ntb);
    SeAM(c,ntc);
    printf("%d\n",dfs(0,0,0));
    return 0;
}
```

跟例3的一样的水题

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=150+5,M=1e6+5,inf=0x3f3f3f3f,mod=1e8;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
char a[N],b[N],c[N];
int nta[N][26],ntb[N][26],ntc[N][26];
ll f[N][N][N];
void SeAM(char *a,int nt[][26]){
    int len=strlen(a+1);
    for(int i=len;i;nt[i-1][a[i]-'a']=i,i--)
        memcpy(nt[i-1],nt[i],sizeof nt[i]);
}
ll dfs(int x,int y,int z){
    if(f[x][y][z]) return f[x][y][z];
    //printf("(%d,%d)\n",x,y);
    for(int i=0;i<26;i++){
        if(nta[x][i]&&ntb[y][i]&&ntc[z][i])
        f[x][y][z]=f[x][y][z]+dfs(nta[x][i],ntb[y][i],ntc[z][i]);
    }
    if(x&&y&&z) ++f[x][y][z];
    return f[x][y][z];
}
int main(){
    scanf("%s%s%s",a+1,b+1,c+1);
    SeAM(a,nta);
    SeAM(b,ntb);
    SeAM(c,ntc);
    printf("%lld\n",dfs(0,0,0));
    return 0;
}
```

待补……