牛客多校第四场

崩盘了……单调栈写不出来，拉倒（此为赛后第一时间的感想）

行吧。本场的题都很困难，也就只有两道签到题有点意思，别的实在是鸡肋……

这个页面不太想写了，因为太难的题罗列上来对能力提高也没什么帮助。

F

两道签到题，有趣的F题题解已经写在周报里了。

B

B题只是用到了快速幂（不用也罢）和线性筛，代码如下：

```
#include<stdio.h>

#define MOD 1000000007

int prime[1000010]={0};
int f[1000010]={0};

void init()
{
    int i;
    for(i=2;i<=1000005;++i)
    {
        if(!prime[i])
        {
            prime[++prime[0]]=i;
            f[i]=1;
        }
        int j;
        for(j=1;j<=prime[0];j++)
        {
            if(i*prime[j]>1000005)
            {
                break;
            }
            prime[i*prime[j]]=1;
            f[i*prime[j]]=f[i]+1;
            if(i%prime[j]==0)
            {
                break;
            }
```

```
        }
    }
    return;
}

long long QPow(long long bas,long long t)
{
    long long ret=1;
    for(;t;t>>=1,bas=(bas*bas)%MOD)
    {
        if(t&1LL)
        {
            ret=(ret*bas)%MOD;
        }
    }
    return ret;
}

int main()
{
    int t;
    scanf("%d",&t);
    init();
    while(t--)
    {
        long long x,c;
        scanf("%lld%lld",&x,&c);
        long long ans=0;
        long long tmp=QPow(c,(long long)f[x]);
        printf("%lld\n",tmp);
    }
    return 0;
}
```

C

以下是补题。

```
#include<stdio.h>
#include<string.h>

struct Sam
{
    struct Sam *fa, *go[10];
    int val;
};
```

```
void clear(struct Sam ss)
{
    ss.fa = 0;
    ss.val = 0;
    memset(ss.go, 0, sizeof(ss.go));
}

struct Sam *now, *root, *last, *cur, Pool[100010 * 10 * 2], *a[100010];

char s[100010];
int pos[100010], ne[100010];

void Prepare()
{
    cur = Pool;
    clear(*cur);
    root = last = cur;
}

struct Sam *Insert(struct Sam *last, int now)
{
    struct Sam *p = last;
    if(p -> go[now])
    {
        struct Sam *q = p -> go[now];
        if(q -> val == p -> val + 1)return q;
        struct Sam *nt = ++cur;
        clear(*nt);
        nt -> val = p -> val + 1;
        memcpy(nt -> go, q -> go, sizeof(q -> go));
        nt -> fa = q -> fa;
        q -> fa = nt;
        while(p && p -> go[now] == q)p -> go[now] = nt, p = p -> fa;
        return nt;
    }
    struct Sam *np = ++cur;
    clear(*np);
    np -> val = p -> val + 1;
    while(p && !p -> go[now])p -> go[now] = np, p = p -> fa;
    if(!p)np -> fa = root;
    else
    {
        struct Sam *q = p -> go[now];
        if(q -> val == p -> val + 1)
        {
            np -> fa = q;
        }
        else
        {
            struct Sam *nt = ++cur;
            clear(*nt);
```

```
            nt -> val = p -> val + 1;
            memcpy(nt -> go, q -> go, sizeof q -> go);
            nt -> fa = q -> fa;
            q -> fa = nt;
            np -> fa = nt;
            while(p && p -> go[now] == q)p -> go[now] = nt, p = p -> fa;
        }
    }
    return np;
}

int main()
{
    scanf("%s", s + 1);
    int n = strlen(s + 1);
    int i;
    for(i = 0; i < 10; i++)
    {
        pos[i] = n + 1;
    }
    for(i = n; i; i--)
    {
        ne[i] = n + 1;
        int j;
        for(j = s[i] - 'a'; j < 10; j++)
        {
            ne[i] =(ne[i]<pos[j]?ne[i]:pos[j]);
        }
        pos[s[i] - 'a'] = i;
    }
    Prepare();
    a[n + 1] = last;
    for(i = n; i; i--)
    {
        struct Sam *last = a[ne[i]];
        int j;
        for(j = ne[i] - 1; j >= i; j--)
        {
            last = Insert(last, s[i] - 'a');
        }
        a[i] = last;
    }
    long long ans = 0;
    struct Sam *now;
    for(now = cur; now > Pool; now--)
    {
        ans += now -> val - now -> fa -> val;
    }
    printf("%lld\n", ans);
```

```
}
```