

# 数论 1

## 逆元递推

### 算法简介

线性时间递推  $0 \leq i < p-1$  在模  $p$  意义下的乘法逆元。

### 算法思想

首先  $1^{-1} \equiv 1 \pmod p$

设  $p = k \cdot i + r \ (1 \leq r < i < p)$

所以有  $k \cdot i + r \equiv 0 \pmod p$

两边同乘以  $r^{-1}$  有  $i^{-1} \equiv -k \cdot r^{-1} \pmod p$

即  $i^{-1} \equiv -\lfloor \frac{p}{i} \rfloor \cdot i^{-1} \pmod p$

### 算法模板

[洛谷p3811](#)

```
const int MAXP=3e6+5;
int inv[MAXP];
void get_inv(int p){
    inv[1]=1;
    for(i,2,p)
        inv[i]=1LL*(p-p/i)*inv[p%i]%p;
}
```

## 数论分块

### 算法简介

数论分块用于解决形如  $\sum_{i=1}^n a_{\lfloor \frac{n}{i} \rfloor}$  的问题，通过预处理  $a_i$  前缀和，可以把时间复杂度降到  $O(\sqrt{n})$

### 算法思想

由于部分  $\lfloor \frac{n}{i} \rfloor$  的值相同，所以考虑分块计算。

设对  $\forall i \in \lfloor l, r \rfloor$  有  $\lfloor \frac{n}{i} \rfloor$  的值相同。

显然  $i$  等于上一个块的  $r$  加  $1$ ，故只需要考虑  $r$  的计算。

事实上有 
$$\lfloor \frac{n}{\lfloor \frac{n}{\lfloor \frac{n}{r} \rfloor} \rfloor} \rfloor \leq \frac{n}{\lfloor \frac{n}{\lfloor \frac{n}{r} \rfloor} \rfloor + 1}$$

移项，得 
$$\frac{n}{\lfloor \frac{n}{\lfloor \frac{n}{r} \rfloor} \rfloor + 1} \leq \lfloor \frac{n}{\lfloor \frac{n}{r} \rfloor} \rfloor$$

所以有 
$$r = \lfloor \frac{n}{\lfloor \frac{n}{\lfloor \frac{n}{r} \rfloor} \rfloor} \rfloor$$

关于算法的时间复杂度分析，有时间复杂度等于  $\lfloor \frac{n}{i} \rfloor$  的可能取值个数。

当  $\lfloor \frac{n}{i} \rfloor \leq \sqrt{n}$  时，显然这样的取值不超过  $\sqrt{n}$  个。

当  $\lfloor \frac{n}{i} \rfloor > \sqrt{n}$  时，有  $i < \sqrt{n}$  所以这样的取值也不超过  $\sqrt{n}$  个。

综上所述  $\lfloor \frac{n}{i} \rfloor$  的可能取值个数只有  $O(\sqrt{n})$  个，时间复杂度证毕。

## 算法模板

[洛谷p2261](#)

### 题意

给定  $n$  和  $k$  请计算  $\sum_{i=1}^n k \lfloor \frac{n}{i} \rfloor \pmod i$

### 题解

$$\sum_{i=1}^n k \lfloor \frac{n}{i} \rfloor \pmod i = \sum_{i=1}^n k - i \lfloor \frac{n}{i} \rfloor = n \lfloor \frac{n}{1} \rfloor - \sum_{i=1}^n i \lfloor \frac{n}{i} \rfloor$$

```
LL pre_s(int k){return 1LL*k*(k+1)/2;}
LL cal(int i,int n){
    LL ans=0;
    int lef=1,rig=0;
    while(lef<=i){
        rig=min(i,n/(n/lef));
        ans+=(pre_s(rig)-pre_s(lef-1))*(n/lef);
        lef=rig+1;
    }
    return ans;
}
int main()
{
    LL n=read_int(),k=read_int();
```

```

    enter(n*k-cal(min(n,k),k));
    return 0;
}

```

## 算法练习

[洛谷p2260](#)

### 题意

求 
$$\sum_{i=1}^n \sum_{j=1}^m [i \neq j] (n \bmod i) \ast (m \bmod j) \pmod{19940417}$$

### 题解

不妨设  $n \leq m$

$$\sum_{i=1}^n \sum_{j=1}^m [i \neq j] (n \bmod i) \ast (m \bmod j) = \sum_{i=1}^n (n \bmod i) \ast \sum_{j=1}^m (m \bmod j) - \sum_{i=1}^n (n \bmod i) \ast (m \bmod i)$$

前面两项解法同模板题，主要关注最后一项。

$$\sum_{i=1}^n (n \bmod i) \ast (m \bmod i) = \sum_{i=1}^n n \ast m - n \ast i \ast \lfloor \frac{m}{i} \rfloor - m \ast i \ast \lfloor \frac{n}{i} \rfloor + i^2 \ast \lfloor \frac{n}{i} \rfloor \ast \lfloor \frac{m}{i} \rfloor$$

类似的，也可以用数论分块解决，注意这条式子的最后一项需要同时处理  $n$  和  $m$  的分块边界。

```

const int mod=19940417;
LL inv6;
LL exgcd(LL a,LL b,LL &tx,LL &ty){
    if(b==0){
        tx=1,ty=0;
        return a;
    }
    LL re=exgcd(b,a%b,ty,tx);
    ty-=a/b*tx;
    return re;
}
LL pre_s_1(int k){return 1LL*k*(k+1)/2%mod;}
LL s_1(int lef,int rig){return (pre_s_1(rig)-pre_s_1(lef-1))%mod;}
LL pre_s_2(int k){return 1LL*k*(k+1)%mod*(2*k+1)%mod*inv6%mod;}
LL s_2(int lef,int rig){return (pre_s_2(rig)-pre_s_2(lef-1))%mod;}
LL cal_1(int i,int n){
    LL ans=0;
    int lef=1,rig=0;
    while(lef<=i){

```

```
        rig=min(i,n/(n/lef));
        ans=(ans+s_1(lef,rig)*(n/lef)%mod)%mod;
        lef=rig+1;
    }
    return ans;
}
LL cal_2(int i,int n,int m){
    LL ans=0;
    int lef=1,rig=0;
    while(lef<=i){
        rig=min(i,min(n/(n/lef),m/(m/lef)));
        ans=(ans+s_2(lef,rig)*(n/lef)%mod*(m/lef)%mod)%mod;
        lef=rig+1;
    }
    return ans;
}
int main()
{
    LL n=read_int(),m=read_int(),ans,t;
    if(n>m)
        swap(n,m);
    exgcd(6,mod,inv6,t);
    inv6%=mod;
    ans=(n*n-cal_1(n,n))%mod*((m*m-cal_1(m,m))%mod)%mod;
    ans=(ans-n*n%mod*m%mod)%mod;
    ans=(ans+n*cal_1(n,m)%mod)%mod;
    ans=(ans+m*cal_1(n,n)%mod)%mod;
    ans=(ans-cal_2(n,n,m))%mod;
    ans=(ans+mod)%mod;
    enter(ans);
    return 0;
}
```

From:  
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal\\_string:jxm2001:%E6%95%B0%E8%AE%BA\\_1&rev=1595738397](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E6%95%B0%E8%AE%BA_1&rev=1595738397)

Last update: 2020/07/26 12:39