数论 1

逆元递推

算法简介

线性时间递推 $0\sim p-1$ 在模 p 意义下的乘法逆元。

算法思想

首先 $1^{-1}\equiv 1\pmod p$□

设 $p=k\ast i+r\left(1\le r \lt i\lt p\right)$□

所以有 $k\ast i+r\equiv 0\pmod p$□

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

即 $i^{-1}\equiv -\lfloor \frac pi\rfloor\ast (p\ \text{mod}\ i)^{-1}\pmod p$

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 $\sum_{i=1}^n a_if(\lfloor \frac ni\rfloor)$ 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 $\lfloor \frac ni\rfloor$ 的值相同，所以考虑分块计算。

设对 $\forall i \in\lfloor l, r\rfloor$□有 $\lfloor \frac ni\rfloor$ 的值相同。

显然 l 等于上一个块的 r 加 1，故只需要考虑 r 的计算。

事实上有 \begin{equation}\lfloor \frac n{\lfloor \frac nl\rfloor}\rfloor\le \frac n{\lfloor \frac nl\rfloor}\lt \lfloor \frac n{\lfloor \frac nl\rfloor}\rfloor+1\end{equation}

移项，得 \begin{equation}\frac n{\lfloor \frac n{\lfloor \frac nl\rfloor}\rfloor+1}\lt\lfloor \frac nl\rfloor\le \frac n{\lfloor \frac n{\lfloor \frac nl\rfloor}\rfloor}\end{equation}

所以有 \begin{equation}r=\lfloor \frac n{\lfloor \frac nl\rfloor}\rfloor\end{equation}

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

当 $\lfloor \frac ni\rfloor \le \sqrt n$ 时，显然这样的取值不超过 $\sqrt n$ 个。

当 $\lfloor \frac ni\rfloor \gt \sqrt n$ 时，有 $i \lt \sqrt n$□所以这样的取值也不超过 $\sqrt n$ 个。

综上所述□$\lfloor \frac ni\rfloor$ 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 $\sum_{i=1}^n k\ mod \ i$

题解

\begin{equation}\sum_{i=1}^n k\ mod \ i=\sum_{i=1}^n k-i\ast\lfloor \frac ki\rfloor=n\ast k-\sum_{i=1}^n i\ast\lfloor \frac ki\rfloor\end{equation}

```
#include <cstdio>
#include <iostream>
#include <vector>
#include <algorithm>
#include <cstring>
#include <cctype>
#include <queue>
#include <cmath>
#define _for(i,a,b) for(int i=(a);i<(b);++i)
#define _rep(i,a,b) for(int i=(a);i<=(b);++i)
#define mem(a,b) memset((a),(b),sizeof(a))
using namespace std;
typedef long long LL;
inline int read_int(){
```

```
    int t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline LL read_LL(){
    LL t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline void write(LL x){
    register char c[21],len=0;
    if(!x)return putchar('0'),void();
    if(x<0)x=-x,putchar('-');
    while(x)c[++len]=x%10,x/=10;
    while(len)putchar(c[len--]+48);
}
inline void space(LL x){write(x),putchar(' ');}
inline void enter(LL x){write(x),putchar('\n');}
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 \begin{equation}\sum_{i=1}^n\sum_{j=1}^m [i\ne j](n\ mod\ i)\ast (m\ mod\ j)\pmod {19940417}\end{equation}

题解

不妨设 $n\le m$

\begin{equation}\sum_{i=1}^n\sum_{j=1}^m [i\ne j](n\ mod\ i)\ast (m\ mod\ j)=\sum_{i=1}^n (n\ mod\ i)\ast\sum_{j=1}^m (m\ mod\ j)-\sum_{i=1}^n (n\ mod\ i)\ast (m\ mod\ i)\end{equation}

前面两项解法同模板题，主要关注最后一项。

\begin{equation}\sum_{i=1}^n (n\ mod\ i)\ast (m\ mod\ i)=\sum_{i=1}^n n\ast m-n\ast i\ast \lfloor\frac mi\rfloor-m\ast i\ast \lfloor\frac ni\rfloor+i^2\ast\lfloor\frac ni\rfloor\ast\lfloor \frac mi\rfloor\end{equation}

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。

```
#include <cstdio>
#include <iostream>
#include <vector>
#include <algorithm>
#include <cstring>
#include <cctype>
#include <queue>
#include <cmath>
#define _for(i,a,b) for(int i=(a);i<(b);++i)
#define _rep(i,a,b) for(int i=(a);i<=(b);++i)
#define mem(a,b) memset((a),(b),sizeof(a))
using namespace std;
typedef long long LL;
inline int read_int(){
    int t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline LL read_LL(){
    LL t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline void write(LL x){
    register char c[21],len=0;
    if(!x)return putchar('0'),void();
    if(x<0)x=-x,putchar('-');
    while(x)c[++len]=x%10,x/=10;
    while(len)putchar(c[len--]+48);
}
inline void space(LL x){write(x),putchar(' ');}
inline void enter(LL x){write(x),putchar('\n');}
```

```
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;
```

```
}
```