# 前置芝士

## 数论函数

$$\begin{cases} \text{单位函数}\varepsilon = \begin{cases} 1, & n = 1 \\ 0, & otherwise \end{cases} \\ \text{幂函数}Id_k(n) = n^k, \text{当}k = 1\text{时为恒等函数}Id(n)\text{，当}k = 1\text{时为常函数} \\ \text{除数函数}\sigma_k(n) = \sum_{d|n} d^k\text{，当}k = 1\text{时为因数和函数}\sigma(n)\text{，当}k = 0\text{时为因数个数函数}\sigma_o\text{（}n\text{）} \\ \text{欧拉函数}\varphi(n) \end{cases}$$

上述函数均为积性函数，满足$f(1) = 1$，且当$a, b$互质时，有$f(a)f(b) = f(ab)$

## 数论分块

可以快速计算含有除法向下取整的和式。

如果可以在$O(1)$内算出$f(r) - f(l)$或已经预处理出$f$的前缀和时，数论分块可以在$O(\sqrt{n})$的时间内计算上述和式的值。

### 富比尼定理

引理1：$\forall a, b, c \in \mathbb{Z}, [\frac{a}{bc}] = [\frac{[\frac{a}{b}]}{c}]$

引理2：$\forall n \in \mathbb{N}_+, |\{[\frac{n}{d}]|d \in \mathbb{N}_+, d \le n\}| \le [2\sqrt{n}]$

### 数论分块结论

对于常数n，使得式子$[\frac{n}{i}]=[\frac{n}{j}]$

成立的最大的满足的$i\le j\le n$的$j$的值为$[\frac{n}{[\frac{n}{i}]}]$,即$[\frac{n}{i}]$所在的块的右端点为$[\frac{n}{[\frac{n}{i}]}]$

[UVa11526 H(n)]()

```
int H(int n){
    int res=0,l=1,r;
    while(l<=n){
        r=n/(n/l);
        res+=(r-l+1)*(n/l);//原式为res=(res+n/i)
        l=r+1;//处理一些连续的块
    }
    return res;
}
```

[CQOI2007 余数求和]()

计算$G(n,k)=\sum_{i=1}^{n}k \mod i$

```
int G(int n,int k){
    int res=n*k,l=1,r;
    while(l<=n){
        if(k/l!=0) r=min(k/(k/l),n);
        else r=n;
        res-=(r-l+1)*(k/l)*(l+r)/2;
        l=r+1;
    }
    return res;
}
```

## 狄利克雷卷积

定义两个数论函数$f,g$的Dirichlet卷积为$(f*g)(n)=\sum_{d|n}f(d)g(\frac{n}{d})$

对于卷积前n项：$h[i]=\sum_{d|i}f[d]*g[\frac{i}{d}]$

```
F(i,1,n) h[i]=0;
F(i,1,n){
    F(j,1,n/i) h[i*j]=(h[i*j]+f[i]*g[j]%mod)%mod;
}
F(i,1,n) printf("%d ",h[i]);
```

### 常用的结论

- $\varphi*1=id$
- $\mu*1=\epsilon$
- $\mu*id=\varphi$
- $1*1=d$（因数个数）

- $id*1=\sigma$(因数和)

# 莫比乌斯反演

## 定理

$$F(n)\text{和}f(n)\text{是定义在非负整数集合上的两个函数，}$$
$$\text{并且满足条件}F(n)=\sum_{d|n}f(d)\text{，那么我们得到结论：}$$
$$f(n)=\sum_{d|n}\mu(d)F(\frac{n}{d})$$
$$\text{其中莫比乌斯函数}\mu(d)\text{的定义：}$$
$$\mu(n)=\begin{cases}1,n=1\\0,n\text{含有平方因子}\\(-1)^k\text{，}k\text{为}n\text{的本质不同质因子个数}\end{cases}$$

## 性质

(1)对于任意正整数$n$，有$\sum_{d|n}\mu(d)=\begin{cases}1,n=1\\0,n>1\end{cases}$

(2)对于任意正整数$n$，有$\sum_{d|n}\frac{\mu(d)}{d}=\frac{\varphi(n)}{n}$

#### 莫比乌斯变换

如果有$f(n)=\sum_{d|n}g(d)$, 那么有$g(n)=\sum_{d|n}\mu(d)f(\frac{n}{d})$

这种形式下f(n)成为数论g(n)的莫比乌斯变换，数论函数g(n)成为f(n)的莫比乌斯逆变换（反演）

#### 常用结论

①$[gcd(i,j)=1]=\sum_{d|gcd(i,j)}\mu(d)$我们也可以简记为$\sum_{d|n}\mu(d)=[n=1]$

②$d(i*j)=\sum_{x|i}\sum_{y|j}[gcd(x,y)=1]$, 其中$d(i)$为$i$的约数个数

③$\sum_{i=1}^{n}\sum_{j=1}^{m}[gcd(i,j)=k]=\sum_{i=1}^{[\frac{n}{k}]}\sum_{j=1}^{[\frac{m}{k}]}[gcd(i,j)=1]$

$$\sum_{i=1}^{n}\sum_{j=1}^{m}[gcd(i,j)=1](n<m)$$
$$=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{d|gcd(i,j)}\mu(d)$$
$$=\sum_{d=1}^{n}\mu(d)*[\frac{n}{d}]*[\frac{m}{d}]\text{, 可以数论分块}$$

④$\sum_{i=1}^{n}\sum_{j=1}^{m}lcm(i,j)=\sum_{d=1}^{n}d\sum_{k=1}^{[\frac{n}{d}]}\mu(k)*k^2\sum_{i=1}^{[\frac{n}{dk}]}i\sum_{j=1}^{[\frac{m}{dk}]}j$

#### 线性筛求莫比乌斯反演函数

```
void init(){
    memset(vis,0,sizeof(vis));
    mu[1]=1;
    cnt=0;
    for(int i=2;i<N;i++){
        if(!vis[i]) prime[cnt++]=i;
        mu[i]=-1;
    }
    for(int j=0;j<cnt&&i*prime[j]<N;j++){
        vis[i*prime[j]]=1;
        if(i%prime[j]) mu[i*prime[j]]=-mu[i];
        else{
            mu[i*prime[j]]=0;
            break;
        }
    }
}
```

#### 证明莫比乌斯反演定理

$\sum_{d|n}\mu(d)F(\frac{n}{d})=\sum_{d|n}\mu(d)\sum_{d'|\frac{n}{d}}f(d')=\sum_{d'|n}f(d')\sum_{d|\frac{n}{d'}}\mu(d)=f(n)$

#### luogu P2522 [HAOI2011]Problem b

求值$\sum_{i=x}^{n}\sum_{j=y}^{m}[gcd(i,j)=k]$

可容斥分为四块，每块化简为$\sum_{d=1}^{min([\frac{n}{k}],[\frac{m}{k}])}\mu(d)[\frac{n}{kd}][\frac{m}{kd}]$

时间复杂度$O(N+T\sqrt{n})$

```
#include<bits/stdc++.h>
#define inf 0x3f3f3f3f
#define int long long
using namespace std;
const int N=1e6+7;

int is_pri[N],pri[N],mu[N],phi[N],sum[N],cnt;

void init(){ //预处理
    memset(is_pri,1,sizeof(is_pri));
    mu[1]=1;phi[1]=1;sum[1]=1;
    cnt=0;
    for(int i=2;i<N;i++){
        if(is_pri[i]){
            pri[++cnt]=i;
            phi[i]=i-1;
            mu[i]=-1;
        }
        for(int j=1;j<=cnt&&i*pri[j]<N;j++){
            is_pri[i*pri[j]]=0;
            if(i%pri[j]){
                mu[i*pri[j]]=-mu[i];
```

```
                phi[i*pri[j]]=phi[i]*(pri[j]-1);
            }else{
                phi[i*pri[j]]=phi[i]*pri[j];
                mu[i*pri[j]]=0;
                break;
            }
        }
        sum[i]=sum[i-1]+mu[i]; //前缀和
    }
}

int F(int n,int m,int k){
    int res=0;
    /*数论分块的原型
    for(int d=1;d<=min(n,m);d++){
        res+=mu[d]*(n/d)*(m/d);
    }*/
    for(int l=1,r;l<=min(n,m);l=r+1){
        r=min(n/(n/l),m/(m/l));
        res+=(sum[r]-sum[l-1])*(n/l)*(m/l);
    }
    return res;
}

signed main(){
    init();
    int n,a,b,c,d,k,ans;
    cin>>n;
    while(n--){
        cin>>a>>b>>c>>d>>k;
        ans=F(b/k,d/k,k)-F((a-1)/k,d/k,k)-F(b/k,(c-1)/k,k)+F((a-1)/k,(c-
1)/k,k);//容斥
        cout<<ans<<"\n";
    }
    return 0;
}
```

**luogu P1829 [国家集训队]Crash的数字表格**

求值$\sum_{i=1}^{n}\sum_{j=1}^{m}\operatorname{lcm}(i,j) \qquad (n,m\leqslant 10^7)$

$$\text{原式等于} \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{i\cdot j}{\gcd(i,j)}$$

$$=\sum_{d=1}^{n} d\cdot\sum_{i=1}^{\lfloor\frac{n}{d}\rfloor}\sum_{j=1}^{\lfloor\frac{m}{d}\rfloor}[\gcd(i,j)=1]\ i\cdot j$$

$$=\sum_{d=1}^{n}\sum_{d|i}^{n}\sum_{d|j}^{m}\mu(d)\cdot i\cdot j$$

$$\text{令}g(n,m)=\sum_{i=1}^{n}\sum_{j=1}^{m} i\cdot j=\frac{n\cdot(n+1)}{2}\times\frac{m\cdot(m+1)}{2}$$

$$\operatorname{sum}(n,m)=\sum_{d=1}^{n}\mu(d)\cdot d^2\cdot g(\lfloor\frac{n}{d}\rfloor,\lfloor\frac{m}{d}\rfloor)$$

$$\text{原式}=\sum_{d=1}^{n} d\cdot\operatorname{sum}(\lfloor\frac{n}{d}\rfloor,\lfloor\frac{m}{d}\rfloor),\text{可用数论分块和线性筛解决。}$$

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int N=1e7+7;
const int mod=20101009;

int np[N],pri[N],mu[N],sum[N],cnt=0;
void init(){
    mu[1]=np[1]=1;
    for(int i=2;i<N;++i){
        if(!np[i])  pri[++cnt]=i,mu[i]=-1;
        for(int j=1;j<=cnt&&pri[j]*i<N;++j){
            np[pri[j]*i]=1;
            if(i%pri[j]) mu[i*pri[j]]=-mu[i];
            else break;
        }
    }
    for(int i=1;i<N;++i) sum[i]=(sum[i-1]+i*i%mod*(mu[i]+mod)%mod)%mod;
}

int Sum(int x,int y){
    return (x*(x+1)/2%mod)*(y*(y+1)/2%mod)%mod;
}

int func(int x,int y){
    int res=0;
    for(int l=1,r;l<=min(x,y);l=r+1){
        r=min(x/(x/l),y/(y/l));
        res=(res+(sum[r]-sum[l-1]+mod)*Sum(x/l,y/l)%mod)%mod;
    }
    return res;
}

int solve(int x,int y){
    int res=0;
    for(int l=1,r;l<=min(x,y);l=r+1){
        r=min(x/(x/l),y/(y/l));
        res=(res+(r-l+1)*(l+r)/2%mod*func(x/l,y/l)%mod)%mod;
    }
```

```
    return res;
}

signed main(){
    ios::sync_with_stdio(0);
    cin.tie(0);cout.tie(0);
    init();
    int n,m;
    cin>>n>>m;
    cout<<solve(n,m)<<"\n";
    return 0;
}
```

**luoguP3327 [SDOI2015]约数个数和**

求 $\sum_{i=1}^{n}\sum_{j=1}^{m} d(i\cdot j)\left(d(n)=\sum_{i|n}1\right) n,m,T\leq 5\times 10^4$

其中d(n)表示n的约数个数。

$$\begin{aligned}
d(i\cdot j) &= \sum_{x|i}\sum_{y|j}[\gcd(x,y)=1] \\
&= \sum_{x|i}\sum_{y|j}\sum_{p|\gcd(x,y)}\mu(p) \\
&= \sum_{p=1}^{min(i,j)}\sum_{x|i}\sum_{y|j}[p\mid\gcd(x,y)]\cdot\mu(p) \\
&= \sum_{p|i,p|j}\mu(p)\sum_{x|i}\sum_{y|j}[p\mid\gcd(x,y)] \\
&= \sum_{p|i,p|j}\mu(p)\sum_{x|\frac{i}{p}}\sum_{y|\frac{j}{p}}1 \\
&= \sum_{p|i,p|j}\mu(p)d\left(\frac{i}{p}\right)d\left(\frac{j}{p}\right)
\end{aligned}$$

将上述式子代回原式

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{m}d(i\cdot j) \\
=&\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{p|i,p|j}\mu(p)d\left(\frac{i}{p}\right)d\left(\frac{j}{p}\right) \\
=&\sum_{p=1}^{min(n,m)}\sum_{i=1}^{n}\sum_{j=1}^{m}[p\mid i,p\mid j]\cdot\mu(p)d\left(\frac{i}{p}\right)d\left(\frac{j}{p}\right) \\
=&\sum_{p=1}^{min(n,m)}\sum_{i=1}^{\left\lfloor\frac{n}{p}\right\rfloor}\sum_{j=1}^{\left\lfloor\frac{m}{p}\right\rfloor}\mu(p)d(i)d(j) \\
=&\sum_{p=1}^{min(n,m)}\mu(p)\sum_{i=1}^{\left\lfloor\frac{n}{p}\right\rfloor}d(i)\sum_{j=1}^{\left\lfloor\frac{m}{p}\right\rfloor}d(j) \\
=&\sum_{p=1}^{min(n,m)}\mu(p)S\left(\left\lfloor\frac{n}{p}\right\rfloor\right)S\left(\left\lfloor\frac{m}{p}\right\rfloor\right)\left(S(n)=\sum_{i=1}^{n}d(i)\right)
\end{aligned}$$

那么$O(n)$预处理前缀和，$O(\sqrt{n})$分块处理询问，就可以解决问题。

```
#include<bits/stdc++.h>
#define inf 0x3f3f3f3f
#define int long long
using namespace std;

const int N=1e5+7;
int is_pri[N],pri[N],g[N],mu[N],sum[N],d[N],t[N],cnt=0;

void init(){
    memset(is_pri,1,sizeof(is_pri));
    mu[1]=1,d[1]=1,is_pri[1]=0;
    for(int i=2;i<N;i++){
        if(is_pri[i]){
            pri[++cnt]=i;
            mu[i]=-1;
            d[i]=2;
            t[i]=1;
        }
        for(int j=1;j<=cnt&&pri[j]*i<N;j++){
            is_pri[pri[j]*i]=0;
            if(i%pri[j]){
                mu[i*pri[j]]=-mu[i];
                t[pri[j]*i]=1; //最小质因子出现的次数
                d[pri[j]*i]=d[i]<<1;//因子个数
            }else{
                t[pri[j]*i]=t[i]+1;
                d[pri[j]*i]=d[i]/(t[i]+1)*(t[i]+2);
                mu[i*pri[j]]=0;
                break;
            }
        }
    }
    for(int i=2;i<N;i++) d[i]+=d[i-1],mu[i]+=mu[i-1];
}

int solve(int n,int m){
    int res=0;
    for(int l=1,r;l<=min(n,m);l=r+1){
        r=min(n/(n/l),m/(m/l));
        res+=(mu[r]-mu[l-1])*d[n/l]*d[m/l];
    }
    return res;
}

signed main(){
    init();
    int n,m,T;
    cin>>T;
    while(T--){
        cin>>n>>m;
        cout<<solve(n,m)<<"\n";
    }
    return 0;
}
```

**luoguP2257 YY的GCD**

给定$n,m$，求$1 \le x \le N, 1 \le y \le M$且$gcd(x,y)$为质数的$(x,y)$有多少对

$$Ans = \sum_{i=1}^{n}\sum_{j=1}^{m}[gcd(i,j) = prime] = \sum_{k \in prime}^{n}\sum_{i=1}^{[\frac{n}{k}]}\sum_{j=1}^{[\frac{m}{k}]}[gcd(i,j) = 1]$$

$$\sum_{k \in prime}^{n}\sum_{i=1}^{[\frac{n}{k}]}\sum_{j=1}^{[\frac{m}{k}]}\sum_{d|gcd(i,j)}\mu(d) = \sum_{k=1}^{n}\sum_{d=1}^{\frac{n}{k}}\mu(d) * [\frac{n}{kd}] * [\frac{m}{kd}](k \in prime)$$

优化1：用T替换kd然后进行预处理，时间复杂度接近$O(n)$

$$Ans = \sum_{T=1}^{n}[\frac{n}{T}] * [\frac{m}{T}]\sum_{k|T,k \in prime}\mu(\frac{T}{k})$$

```
#include<bits/stdc++.h>
using namespace std;
const int N=1e7+7;
typedef long long ll;
int mu[N],pri[N],cnt=0;
ll f[N];
bool np[N];

void init(){
    mu[1]=np[1]=1;
    for(int i=2;i<N;++i){
        if(!np[i]) pri[++cnt]=i,mu[i]=-1;
        for(int j=1;j<=cnt&&pri[j]*i<N;++j){
            np[i*pri[j]]=1;
            if(i%pri[j]) mu[i*pri[j]]=-mu[i];
            else break;
        }
    }
    for(int i=1;i<=cnt;++i){
        for(int j=1;pri[i]*j<N;++j){
            f[j*pri[i]]+=mu[j];
        }
    }
    for(int i=1;i<N;++i) f[i]+=f[i-1]; //\sum_{p|T}\mu_{T/p}
}

ll solve(int n,int m){
    ll res=0;
    for(int l=1,r;l<=n;l=r+1){
        r=min(n/(n/l),m/(m/l));
        res+=(ll)(f[r]-f[l-1])*(ll)(n/l)*(m/l);
    }
    return res;
}

signed main(){
    init();
    int n,m,t;
    scanf("%d",&t);
```

```
    while(t--){
        scanf("%d%d",&n,&m);
        if(n>m) swap(n,m);
        printf("%lld\n",solve(n,m));
    }
    return 0;
}
```

**BZOJ3529 数表**

令$F(i)$为$i$的约数和，$q$次给定$n,m,a$，求：$\sum_{1\le i\le n,1\le j\le m,F(gcd(i,j))\le a} F(gcd(i,j)) \mod 2^{31}$

数据范围$1\le n,m\le 10^5,1\le q\le 2000,1\le a\le 10^9$

令$g(i)$表示$1\le x\le \frac{n}{p},1\le y\le \frac{m}{p},(x,y)=i$的对数，根据之前的经验有
$g(i)=\sum_{i|d}\mu(\frac{d}{i})[\frac{n}{d}][\frac{m}{d}]$,则
$ans=\sum i=1^{min(n,m)}F(i)g(i)=\sum_{i=1}^{min(n,m)}\sum_{i|d}F(i)\mu(\frac{d}{i})[\frac{n}{d}][\frac{m}{d}]$

交换$i$和$d$，仍有$\sum_{d=1}^{min(n,m)}[\frac{n}{d}][\frac{m}{d}]\sum_{i|d}F(i)\mu(\frac{d}{i})$

用$O(nlogn)$预处理出$\sum_{i|d}F(i)\mu(\frac{d}{i})$前缀和即可$O(\sqrt{n})$回答询问。

考虑$F(i)\le a$的限制，离线，对a排序，按照$F(i)$将$i$排序，用树状数组维护即可。时间复杂度$O(nlog^2n+q\sqrt{n}logn)$

```
#include<bits/stdc++.h>
#define mk make_pair
#define F first
#define S second
#define lb(x) (x&(-x))
using namespace std;
typedef pair<int,int>pii;
typedef unsigned long long ull;
const int N=1e5+7;

int cnt=0,mu[N],g[N],pri[N]; //g是因子和
bool np[N];
pii f[N];   //f对应因子和及对应的数

void init(){  //预处理
    mu[1]=np[1]=1;
    f[1]=mk(1,1);
    for(int i=2;i<N;++i){
        if(!np[i]){
            pri[++cnt]=i;mu[i]=-1;
            g[i]=i+1;f[i]=mk(i+1,i);
        }
        for(int j=1;j<=cnt&&i*pri[j]<N;++j){
            np[pri[j]*i]=1;
            if(i%pri[j]==0){
                mu[i*pri[j]]=0;
                g[i*pri[j]]=g[i]*pri[j]+1;
                f[i*pri[j]]=mk(f[i].F/g[i]*g[i*pri[j]],i*pri[j]);
                break;
            }else{
                mu[i*pri[j]]=-mu[i];
```

```
                f[i*pri[j]]=mk(f[i].F*f[pri[j]].F,i*pri[j]);
                g[i*pri[j]]=pri[j]+1;
            }
        }
    }
}

int tr[N<<1];
void update(int x,int w){for(;x<N;x+=lb(x)) tr[x]+=w;}
int query(int x){int res=0;for(;x;x-=lb(x)) res+=tr[x];return res;}

struct Q{
    int n,m,a,id;
    void read(int i){cin>>n>>m>>a;id=i;}
    bool operator <(const Q &B)const{return a<B.a;}
}qry[N];

int ans[N];

int solve(int n,int m){
    if(n>m) swap(n,m);
    int res=0;
    for(int l=1,r;l<=n;l=r+1){  //数论分块
        r=min(n/(n/l),m/(m/l));
        res=(res+(query(r)-query(l-1))*(n/l)*(m/l));
    }
    return res;
}

signed main(){
    ios::sync_with_stdio(0);
    cin.tie(0);cout.tie(0);
    init();
    stable_sort(f+1,f+N);  //按因子和排序再插入
    int T;
    cin>>T;
    for(int i=1;i<=T;++i) qry[i].read(i);
    stable_sort(qry+1,qry+1+T); //离线处理询问，按a排序
    for(int i=1,j=1;i<=T;++i){
        while(f[j].F<=qry[i].a&&j<=N){
            for(int k=f[j].S;k<N;k+=f[j].S) update(k,f[j].F*mu[k/f[j].S]);
            ++j;
        }
        ans[qry[i].id]=solve(qry[i].n,qry[i].m);
    }
    for(int i=1;i<=T;++i) cout<<(ans[i]&(~(1<<31)))<<"\n"; //相当于模2^31
    return 0;
}
```

## 参考资料


OI-wiki

《信息学奥赛之数学一本通》 林厚从

《莫比乌斯反演》 方泓杰

https://blog.csdn.net/zsjzliziyang/article/details/107749294

https://www.cnblogs.com/peng-ym/p/9446555.html

https://www.cnblogs.com/peng-ym/p/8652288.htmlhttps://www.luogu.com.cn/blog/An-Amazing-Blog/mu-bi-wu-si-fan-yan-ji-ge-ji-miao-di-dong-xi