整除分块

```
for(int l=1,r=1;l<=n;l=r+1){
        if(k/l==0) break;
        r=std::min(k/(k/l), n);
        ans-=(k/l)*(l+r)*(r-l+1)/2;
}
```

线性求逆元

```
inv[0]=inv[1]=1;
for(int i=2;i<=n;++i)
    inv[i]=Mod((mod-mod/i)*inv[mod%i],mod);
```

扩展欧几里得

```
int exgcd(int a,int b,int &x,int &y){ //ax+by=d 其中d=(a,b)
    if(!b){
        x=1,y=0;
        return a;
    }
    int d=exgcd(b,a%b,y,x);
    y-=a/b*x;
    return d;
}
```

欧拉筛

```
bool numlist[N];
int prime[N],cnt;
void Eular(int n=100000){
    for(int i=2;i<=n;i++){
        if(!numlist[i])
            prime[++cnt]=i;
        for(int j=1;prime[j]<=n/i;j++){
```

```
            numlist[i*prime[j]] =true;
            if(i%prime[j]==0)
                break;
        }
    }
    return ;
}
```

线性基

给定 nn 个整数（可能重复），现在需要从中挑选任意个整数，使得选出整数的异或和最大。

请问，这个异或和的最大可能值是多少。

```
const int N = 100010;
LL n,m,a[N];

void solve(){
    n=read();
    rep(i,0,n-1) a[i]=read();

    int k=0;
    for(int i=62;i>=0;--i){
        for(int j=k;j<n;++j){
            if(a[j]>>i&1) {
                swap(a[j],a[k]);
                break;
            }
        }
        if(!(a[k]>>i&1)) continue;
        for(int j=0;j<n;++j)
            if(j!=k&&(a[j]>>i&1))
                a[j]^=a[k];
        k++;
        if(k==n) break;
    }
    LL ans=0;
    for(int i=0;i<k;++i) ans^=a[i];
    print(ans);
}
```

给定你由 N 个整数构成的整数序列，你可以从中选取一些（至少一个）进行异或（xor）运算，从而得到很多不同的结果。

请问，所有能得到的不同的结果中第 k小的结果是多少。

```
const int N=20010,M=N*2,mod=1e9+7;
LL n,m,k,a[N];

void solve(int Case){
    printf("Case #%d:\n",Case);
```

```
    n=read();
    rep(i,0,n-1) a[i]=read();
    int k=0;
    for(int i=62;i>=0;--i){
        for(int j=k;j<n;++j){
            if(a[j]>>i&1){
                swap(a[j],a[k]);
                break;
            }
        }
        if(!(a[k]>>i&1)) continue;
        for(int j=0;j<n;++j)
            if(j!=k&&a[j]>>i&1)
                a[j]^=a[k];
        k++;
        if(k==n) break;
    }
    reverse(a,a+k);

    m=read();
    while(m--){
        LL x=read();
        if(k<n) x--;
        if(x>=(1ll<<k)) {puts("-1");continue;}
        LL ans=0;
        for(int i=0;i<=62;++i){
            if(x>>i&1) ans^=a[i];
        }
        print(ans);
    }
}
```

中国剩余定理

```
struct EX_CRT{
    LL n,a[N],b[N]; //x mod a[i] = b[i];
    void read(){
        cin >> n;
        for(int i=1;i<=n;++i) cin>>a[i]>>b[i];
    }
    LL gsm(LL a,LL b,LL mod){
        LL x0=0;
        while(b){
            if(b&1) x0=(x0+a)%mod;
            a=(a+a)%mod;
            b>>=1;
        }
        return x0;
    }
    LL exgcd(LL a,LL b,LL &x,LL &y){
        if(!b){
            x=1,y=0;
```

```
            return a;
        }
        LL d=exgcd(b,a%b,y,x);
        y-=a/b*x;
        return d;
    }
    LL ex_crt(){
        LL x0=b[1],X=a[1],x,y,c,d;
        for(int i=2;i<=n;++i){
            c=(b[i]-x0%a[i]+a[i])%a[i];
            d=exgcd(X, a[i], x, y);
            if(c%d) return -1;
            x=gsm(x, c/d, a[i]/d);
            x0+=x*X;
            X*=a[i]/d;
            x0=(x0%X+X)%X;
        }
        return x0;
    }
}A;
```

bsgs

```
LL bsgs(int a,int b,int p){
    if(1%p==b%p) return 0;
    unordered_map<LL,LL> hash; hash.clear();
    LL k=sqrt(p)+1;
    for(int i=0,j=b%p;i<k;++i){
        hash[j]=i;
        j=1ll*j*a%p;
    }
    LL tmp=fpower(a,k,p);
    for(LL i=1,j=tmp;i<=k;++i){
        if(hash.count(j)) return i*k-hash[j];
        j=j*tmp%p;
    }
    return -1;
}
```

扩展BSGS

```
LL fpower(LL a,LL b,LL mod){
    LL ans=1;
    while(b){
        if(b&1) ans=ans*a%mod;
        a=a*a%mod;
        b>>=1;
    }
    return ans;
}
```

```
LL exgcd(LL a,LL b,LL &x,LL &y){
    if(!b){
        x=1,y=0;
        return a;
    }
    LL d=exgcd(b, a%b, y, x);
    y-=a/b*x;
    return d;
}

LL bsgs(LL a,LL b,LL p){
    if(1%p==b%p) return 0;
    unordered_map<LL,LL> hash; hash.clear();
    LL k=sqrt(p)+1;
    for(LL i=0,j=b%p;i<k;++i){
        hash[j]=i;
        j=j*a%p;
    }
    LL tmp=fpower(a,k,p);
    for(LL i=1,j=tmp;i<=k;++i){
        if(hash.count(j)) return i*k-hash[j];
        j=j*tmp%p;
    }
    return -INF;
}

LL exbsgs(LL a,LL b,LL p){
    b=Mod(b,p);
    if(1%p==b%p) return 0;
    LL x,y;
    LL d=exgcd(a,p,x,y);
    if(d>1){ //gcd>1
        if(b%d) return -INF;
        exgcd(a/d,p/d,x,y);
        return exbsgs(a, b/d*x%(p/d), p/d)+1;
    }
    return bsgs(a,b,p);
}
```

积性函数

定义：

如果函数$f:N\to R$满足于任意一对互质的正整数p,q.都有$f(pq)=f(p)f(q)$则称f为积性函数

命题：

如果$f(n)$坏人$g(n)$为积性函数，则$h(n)=f(n)g(n)$也为积性函数

设f为积性函数，假设$n=p_1^{q_1}\dots p_k^{q_k},f(n)=f(p_1^{q_1})\dots f(p_k^{q_k})$

```
f[1] = 1;
for (int i = 2; i <= n; i++) {
```

```
    if (!not prime[i]) p[++tot] = i, cnt[i] = 1, f[i] = calc_f(i, 1);
    for (int j = 1; j <= tot && i * p[j] <= n; j++) {
        not prime[i * p[j]] = 1;
        if (i % p[j] == 0) {
        cnt[i * p[j]] = cnt[i] + 1;
        f[i * p[j]] = f[i]/calc_f(p[j], cnt[i])*calc_f(p[j], cnt[i]+1);
        break;
        }
        cnt[i * p[j]] = 1
        f[i * p[j]] = f[i] * calc_f(p[j], 1);
    }
}
```

线性筛求n个数的正因子个数

```
const int N=10000010,M=N*2,mod=1e9+7;
int n,m,k;
LL Cnt[N],d[N];
bool numlist[N];
int prime[N],cnt;
void Eular(int n=10000000){
    Cnt[1]=1;
    for(int i=2;i<=n;i++){
        if(!numlist[i]){
            prime[++cnt]=i;
            Cnt[i]=2;
            d[i]=1;
        }
        for(int j=1;prime[j]<=n/i;j++){
            numlist[i*prime[j]] =true;
            if(i%prime[j]==0){
                Cnt[i*prime[j]]=Cnt[i]/(d[i]+1)*(d[i]+2);
                d[i*prime[j]]=d[i]+1;
                break;
            }
            d[i*prime[j]]=1;
            Cnt[i*prime[j]]=Cnt[i]*Cnt[prime[j]];
        }
    }
    return ;
}
```

例题

E. Bash Plays with Functions

很容易观察出$f(x)$是积性函数，因此我们先预处理出所有的$f_r(1-20)$因为1e6以内分解质因数后最多的质数的指数为20.然后logn内分解并可以乘得答案。

莫比乌斯反演

作用：

已知$g(n)$的因数和$f(n)=\sum_{d|n}g(d)$,通过f反求g

实际上只要记住如果能够推到成如$f(n)=\sum_{d|n}g(d)$形式后都能通过反演得到:

$$g(n)=\sum_{d|n}\mu(n/d)f(d)$$

其中$\mu()$为莫比乌斯函数。

例子 环计数问题

假设 n 个元组成一个环，每个元都是 1, 2, ...,**r 中的一个数，两个环是不同的环当且仅当它们不能通过旋转使得两个环中对应的每个元素都相等。求有多少个这样的环。

首先我们构造一个无限长的序列，假设此序列的周期为d，其中d为n的因子。有$r^n=\sum_{d|n}d*f(d)$,因此由反演得$f(n)=\frac{1}{n}\sum_{d|n}\mu(\frac{n}{d})r^d$

反演2

设$f:N\to R$，$g:N\to R$是两个函数，且存在正整数N，对于所有n>N, f(n)=g(n)=0,则:

$f(n)=\sum_{n|m}g(m)$等价于$g(n)=\sum_{n|m}\mu(\frac{m}{n})f(m)$当$m\le N$

例题

两个长度为 n 的序列 a, b，求满足 $gcd(x,y)=1$ 且有$a_{b_x}=b_{a_y}$的对 (x, y) 的数量。$(1\le n\le 10^5,1\le a_i,b_i\le n)$

令$f(d)=\sum_{1\le x,y\le n}[a_{b_x}=b_{a_y}][gcd(x,y)=d]$, $g(d)=\sum_{d|x,d|y}[a_{b_x}=b_{a_y}]$

由反演2得:$f(d)=\sum_{d|d'}g(d')\mu(\frac{d'}{d})$，因此答案为$f(1)=\sum_{d=1}^{n}g(d)\mu(d)$

其中g(d)是可以轻松算出来的，总时间复杂度为$O(nlogn)$

一道非常经典的反演题:

E - Sum of gcd of Tuples (Hard)

一个长度为n的序列$A_1,\dots A_n$,求对于K^N种序列的所有$gcd(A_1,A_2\dots A_n)$的和。

$ans=\sum_{d=1}^{n}d\sum_{1\le A_i\le k}[gcd(A_1\dots A_n)]=d]$

令$f(d)=\sum_{1\le A_i\le k}[gcd(A_1,\dots,A_n)=d]$, $g(d)=\sum_{d|d'}f(d')=\lfloor\frac{k}{d}\rfloor^n$

通过反演的第2种形式:

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

所以最终答案:$ans=\sum_{d=1}^{n}d\cdot f(d)$

$=\sum_{d=1}^{n}\sum_{d|d'}\lfloor\frac{k}{d'}\rfloor$

$=\sum_{d'=1}^{n}\sum_{d|d'}d\cdot\mu(\frac{d'}{d})$

$= \sum_{d'=1}^{n} \lfloor \frac{k}{d'} \rfloor \phi(d')$

迪利克雷卷积

定理：

设$f:N \to R$,$g:N \to R$是两个函数，则它们的迪利克雷卷积为$(f*g)(n)=\sum_{d|n} f(d)g(\frac{n}{d})$

通过迪利克雷卷积可以将莫比乌斯反演写成：

$f=g*1 <==> g=f*\mu$

> 一些公式
>
> $\frac{\phi(n)}{n}=\sum_{d|n}\mu(d)\frac{n}{d}$
>
> 应用：快速求$M(n)=\sum_{i=1}^{n}\mu(i)$，其中$n \le 10^{11}$ (杜教筛)
>
> $M(n)=1-\sum_{i=2}^{n}M(\lfloor \frac{n}{i} \rfloor)$,使用整除分块

广义莫比乌斯反演

设$(X,\le)$是下有限的偏序集，$f,g:X \to R$则$f(x)=\sum_{y \le x} g(y)$等价于$g(x)=\sum_{y \le x}\mu(y,x)f(y)$

其中$\mu(x.y)$满足

$$f(x)=\begin{cases} 1 & x=y \\ -\sum\limits_{x \le z<y}\mu(x,y) & x!=y \end{cases}$$

先对于每个T预处理$\sum\limits_{p|T}\mu(p)\frac{p}{T}$，然后注意到$\sum\limits_{T|A_i}\sum\limits_{T|A_j}A_iA_j=\frac{(\sum\limits_{T|A_i}A_i)^2-\sum\limits_{T|A_i}A_i^2}{2}$，对于每一个$T$枚举倍数算出$\sum\limits_{T|A_i}A_i$和$\sum\limits_{T|A_i}A_i^2$即可。