

首先是杜教筛套路式子

$$\text{求 } \sum_{x=1}^n f(x)$$

其中n很大一般 $10^9 \sim 2^{32}$  左右导致常规线性筛超时

一般做法：

$$\text{令 } S(x) = \sum_{i=1}^x f(i)$$

再找一和  $f$  卷积起来有特殊性质的函数  $g(x)$

$$\begin{aligned} \sum_{i=1}^n f * g(i) &= \sum_{i=1}^n \sum_{d|i} g(d) * f\left(\frac{i}{d}\right) \\ &= \sum_{d=1}^n g(d) * \sum_{d|i} f\left(\frac{i}{d}\right) \\ &= \sum_{d=1}^n g(d) * \sum_{i=1}^{n/d} f(i) \\ &= \sum_{d=1}^n g(d) * S\left(\frac{n}{d}\right) \\ &= \sum_{d=2}^n g(d) * S\left(\frac{n}{d}\right) + g(1) * S(n) \end{aligned}$$

然后就发现右边等式最后一项除以 $g(1)$ 是要求的答案

而  $\sum_{d=2}^n g(d) * S\left(\frac{n}{d}\right)$  这玩意可以递归下去

$\sum_{i=1}^n f * g(i)$  一般会在我们的选择下比较好算

先是一道模板题

## BZOJ3944

### Description

给定一个正整数 $N(N \leq 2^{31} - 1)$ 求  $ans1 = \sum_{i=1}^N \phi(i)$      $ans2 = \sum_{i=1}^N \mu(i)$

### Input

一共 $T + 1$ 行 第1行为数据组数 $T(T \leq 10)$  第2  $T + 1$ 行每行一个非负整数 $N$ ，代表一组询问

## Output

一共T行，每行两个用空格分隔的数ans1,ans2

对于这道题可以知道

$$(\mu * 1) = [x = 1]$$

$$\phi * 1 = x$$

$$\text{令 } g(x) = 1$$

对于  $\mu$

$$\sum_{i=1}^n f * g(i) = 1$$

$$\text{所以 } S(n) = 1 - \sum_{d=2}^n S(n/d)$$

对于  $\phi$

$$\sum_{i=1}^n f * g(i) = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$$

$$\text{所以 } S(n) = \frac{n(n+1)}{2} - \sum_{d=2}^n S(n/d)$$

具体实现我就使用map了.=.

```

inline ll solve(int opt,unsigned x){
    if (x<MAXN) return opt?sum1[x]:sum0[x];
    if (M[opt][x]) return M[opt][x];
    ll ret=opt?1ll:1ll*x*(x+1)>>1;
    for (unsigned i=2,j=2;i<=x;i=j+1){
        j=x/(x/i);
        ret-=solve(opt,x/i)*(j-i+1);
    }
    return M[opt][x]=ret;
}

```

傻逼题讲完了T.T

## BZOJ3930

### Description

我们知道，从区间 $[L,H]$ （ $L$ 和 $H$ 为整数）中选取 $N$ 个整数，总共有 $(H-L+1)^N$ 种方案。小z很好奇这样选出的数的最大公约数的规律，他决定对每种方案选出的 $N$ 个整数都求一次最大公约数，以便进一步研究。然而他很快发现工作量太大了，于是向你寻求帮助。你的任务很简单，小z会告诉你一个整数 $K$ ，你需要回答他最大公约数刚好为 $K$ 的选取方案有多少个。由于方案数较大，你只需要输出其除以1000000007的余数即可。

### Input

输入一行，包含4个空格分开的正整数，依次为 $N$ ， $K$ ， $L$ 和 $H$ 。

### Output

输出一个整数，为所求方案数。

### Sample Input

2 2 2 4

### Sample Output

3

### HINT

样例解释

所有可能的选择方案：(2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4), (4, 2), (4, 3), (4, 4)

其中最大公约数等于2的只有3组：(2, 2), (2, 4), (4, 2)

对于100%的数据， $1 \leq N, K \leq 10^9$ ,  $1 \leq L \leq H \leq 10^9$ ,  $H - L \leq 10^5$

一看这数据范围就懵逼了=.=

先 $L, R$ 都除以 $K$ 就变成求区间 $gcd = 1$ 的方案了

然后看上去要杜教筛不过 $R - L \leq 10^5$ 可以枚举选的 $N$ 个数的 $gcd$

一堆差不超过 $10^5$ 的数的 $gcd$ 也在 $10^5$ 以内（易证）

令  $f(x)$  表示  $gcd$  恰好为  $x$  的方案数  
 $g(x)$  表示  $gcd$  是  $x$  的倍数的方案数可以  $O(1)$  求

$$\text{然后 } g(x) = \sum_{x|d} f(d)$$

通过反演模板  $f(x) = \sum_{x|d} \mu(d) g(d)$

$$\text{得 } f(1) = \sum_{i=1} g(i)$$

---

$g(x)$  怎么求？

区间任选 $N$ 个数使得都是一个数 $x$ 的倍数=.=快速幂搞搞

并且选的这 $N$ 个数不能全部一样

所以选的方案要减去这个区间的数的个数（刚好这么多种所有数都一样的方案）

还有个特判为区间是否包含1，包含1的话有1种全部都是1时的gcd为1

```
inline int getf(int x){
    int lt=(L%x)?L/x+1:L/x,rt=R/x;
    return add(qpow(rt-lt+1,N),-rt+lt-1);
}
int main(){
    pre();
    read(N);read(K);read(L);read(R);
    L=L%K?L/K+1:L/K;R/=K;n=R-L;
    for (int i=1;i<=n;i++)
        ans=add(ans,mul(mu[i],getf(i)));
    if (L<=1 && 1<=R)    ans=add(ans,1);
}
```

## BZOJ3529

### Description

有一张  $n \times m$  的数表，其第  $i$  行第  $j$  列（ $1 \leq i \leq n, 1 \leq j \leq m$ ）的数值为 能同时整除  $i$  和  $j$  的所有自然数之和。给定  $a$ ，计算数表中不大于  $a$  的数之和。

### Input

输入包含多组数据。输入的第一行一个整数  $Q$  表示测试点内的数据组数 接下来  $Q$  行，每行三个整数  $n, m, a$  ( $|a| \leq 10^9$ ) 描述一组数据。  $1 \leq n, m \leq 10^5, 1 \leq Q \leq 2 \times 10^4$

### Output

对每组数据，输出一行一个整数，表示答案模  $2^{31}$  的值。

### Sample Input

2 4 3 10 10 5

### Sample Output

20 148

好像很难搞==不过1e5数据范围可以不用杜教筛 吼啊

写成数学符号就是

$$\begin{aligned} \text{令 } F(x) &= \sum_{i|x}^x i \\ \text{求 } \sum_{i=1}^N \sum_{j=1}^M [F(\gcd(i, j)) \leq a] F(\gcd(i, j)) \end{aligned}$$

啥玩意没听过==

先考虑简化版问题：不考虑a的影响怎么做

首先约数和 $F(x)$ 可以线性筛预处理:

假设 $p$ 是个质数

$$\begin{aligned} F(p) &= p + 1 \\ F(x * p) &= F(x) * F(p) \quad (\gcd(x, p) = 1) \\ F(x * p) &= F(x) / (p^0 + p^1 + \dots + p^n) * (p^0 + p^1 + \dots + p^{n+1}) \quad (\gcd(x, p) = p) \end{aligned}$$

非常好然后我们可以

$$\begin{aligned} \text{令 } g(x) &= \sum_{i=1}^N \sum_{j=1}^M [\gcd(i, j) = x] \\ &= \sum_{i=1}^{N/x} \sum_{j=1}^{M/x} [\gcd(i, j) = 1] \\ &= \sum_{i=1}^{N/x} \sum_{j=1}^{M/x} \sum_{d|\gcd(i, j)} \mu(d) \\ &= \sum_{d=1}^{\min(N/x, M/x)} \mu(d) * \left\lfloor \frac{N}{x * d} \right\rfloor * \left\lfloor \frac{M}{x * d} \right\rfloor \end{aligned}$$

令 $k = x * d$

$$g(x) = \sum_{k=x}^{\min(N,M)} \left[\frac{N}{k}\right] \left[\frac{M}{k}\right] \mu\left(\frac{k}{x}\right)$$

所以答案就是

$$\begin{aligned} & \sum_{i=1}^{\min(N,M)} f(i) * g(i) \\ &= \sum_{k=1}^{\min(N,M)} \left[\frac{N}{k}\right] \left[\frac{M}{k}\right] \sum_{x|k} \mu(x) * f(k/x) \end{aligned}$$

就可以用根号分块和预处理前缀和水了

## 然后考虑怎么加入a的限制

emmm

N,M才 $10^5$ ，可以按 $F(x)$ 排序再将a排序，动态维护下前缀和就好了

动态维护前缀和的数据结构比如说树状数组

模 $2^{31}$ 可以自然溢出和 $(1 \ll 31) - 1$ 取与

然后这题就结束了

```
inline void pre(){
    mu[1]=1;t[1]=1;F[1].xx=1;
    for (int i=2;i<=MAXN;i++){
        if (!vis[i])    prime[++cnt]=i,vis[i]=i,F[i].xx=i+1,t[i]=i+1,mu[i]=-1;
        for (int j=1;j<=cnt && i*prime[j]<=MAXN;j++){
            vis[i*prime[j]]=1;
            if (i%prime[j]) {
                mu[i*prime[j]]=-mu[i];
                F[i*prime[j]].xx=F[i].xx*1+prime[j],t[i*prime[j]]=prime[j]+1;
            }else {
                mu[i*prime[j]]=0;
                t[i*prime[j]]=t[i]*prime[j]+1;
                F[i*prime[j]].xx=F[i].xx/t[i]*t[i*prime[j]];
                break;
            }
        }
    }
    for (int i=1;i<=MAXN;i++)    F[i].yy=i;
}

inline void fadd(int x){
    for (int i=1;i*F[x].yy<=MAXN;i++)
        add(i*F[x].yy,F[x].xx*mu[i]);
}

//这里省略
sort(q+1,q+N+1);    sort(F+1,F+MAXN+1);
for (int i=1,j=0;i<=N;i++){
    while (j<MAXN&&F[j+1].xx<=q[i].a)    fadd(++j);
```

```

n=q[i].n;m=q[i].m;
if (n>m) swap(n,m);
for (int l=1,r=1;l<=n;l=r+1){
    r=min(n/(n/l),m/(m/l));
    ans[q[i].id]+=(n/l)*(m/l)*(query(r)-query(l-1));
}
}

```

## BZOJ4652

### Description

牛牛是一个热爱算法设计的高中生。在他设计的算法中，常常会使用带小数的数进行计算。牛牛认为，如果在  $k$  进制下，一个数的小数部分是纯循环的，那么它就是美的。现在，牛牛想知道：对于已知的十进制数  $n$  和  $m$ ，在  $k$  进制下，有多少个数值上互不相等的纯循环小数，可以用分数  $\frac{x}{y}$  表示，其中

$1 \leq x \leq n, 1 \leq y \leq m$ ，且  $x, y$  是整数。一个数是纯循环的，当且仅当其可以写成以下形式：

$a.\dot{c}_1c_2c_3\dots c_{p-1}\dot{c}_p$  其中， $a$  是一个整数， $p \geq 1$ ；对于  $1 \leq i \leq p$ ， $c_i$  是  $k$  进制下的一位数字。例如，在十进制下， $0.45454545\dots = 0.4\dot{5}$  是纯循环的，它可以用  $\frac{5}{11}, \frac{5}{22}$  等分数表示；在十进制下，

$0.1666666\dots = 0.1\dot{6}$  则不是纯循环的，它可以用  $\frac{1}{6}$  等分数表示。需要特别注意的是，我们认为一个整数是纯循环的，因为它的小数部分可以表示成  $0$  的循环或是  $k-1$  的循环；而一个小数部分非  $0$  的有限小数不是纯循环的。

### Input

只有一行，包含三个十进制数  $N, M, K$  意义如题所述，保证  $1 \leq n \leq 10^9, 1 \leq m \leq 10^9, 2 \leq k \leq 2000$

### Output

一行一个整数，表示满足条件的的美的数的个数。

### Sample Input

2 6 10

### Sample Output

4

### HINT

满足条件的数分别是：  $1/1=1.0000\dots$   $1/3=0.3333\dots$   $2/1=2.0000\dots$   $2/3=0.6666\dots$   $1/1$  和  $2/2$  虽然都是纯循环小数，但因为它们相等，因此只计数一次；同样， $1/3$  和  $2/6$  也只计数一次。



啥，纯循环小数，没听过0.0

首先这题很显然可以看出是要求一些分子分母互质的数的个数(不互质的会重复出现)

先不管那么多式子写起来

$$\sum_{i=1}^n \sum_{j=1}^m [\gcd(i, j) = 1] \left[ \text{分数 } \frac{i}{j} \text{ 在 } k \text{ 进制下纯循环} \right]$$

还有一个可能是对的不会证得就是对于一个纯循环小数 $\frac{a}{b}$  如果 $a \times c$  仍与分母 $b$ 互质（不然就被约分消掉了），那么 $\frac{a \times c}{b}$ 应该仍是个纯循环小数（可能是错的我也不会证），然后就说明了是否为纯循环小数在分子分母互质的情况下与分子关系不大（蛤也不敢说是不是完全没有关系）

还是以1为分子在10进制下找规律打表

$$\begin{aligned}
1/1 &= 1\checkmark \\
1/2 &= 0.5\times \\
1/3 &= 0.\dot{3}\checkmark \\
1/4 &= 0.25\times \\
1/5 &= 0.2\times \\
1/6 &= 0.1\dot{6}\times \\
1/7 &= \text{猫猫猫} = 0.\dot{1}42857\checkmark \\
1/8 &= 0.125\times \\
1/9 &= 0.\dot{1}\checkmark \\
1/10 &= 0.1\times \\
1/11 &= 0.0\dot{9}\checkmark \\
1/12 &= 0.08\dot{3}\times \\
&\dots
\end{aligned}$$

将符合要求的分母写下来1, 3, 7, 9, 11 嗯没有偶数，但也跟质数不像，然后通过严谨而又细致的猜测推理我们发现这东西应该是要求分母与k互质

具体为什么=.=

## 证明From 神仙网友

我们先来回想一下，我们是怎样使用除法来判断一个分数 $\frac{x}{y}$ 是否是纯循环小数的。显然我们是一路除下去，什么时候出现了相同的余数，那么这个数就是一个循环小数。如果第一个重复的余数是 $x$ ，那么这个数就是纯循环小数。这种方法实际上就是存在一个数 $l \neq 0$ ，使得：

$$xk^l \equiv x \pmod{y}$$

又由于题目要求值不重复，那么我们可以得到 $x \perp y$ 。所以，我们可以推出 $k^l \equiv 1 \pmod{y}$  ( $l > 0$ )。所以我们只需 $k \perp y$ 即可。

## 所以事情变成了我们熟悉的样子

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

然后开始乱搞

$$\begin{aligned}
&\sum_{i=1}^n \sum_{j=1}^m [\gcd(i, j) = 1] [\gcd(j, k) = 1] \\
&= \sum_{j=1}^m [\gcd(j, k) = 1] \sum_{i=1}^n [\gcd(i, j) = 1] \\
&= \sum_{j=1}^m [\gcd(j, k) = 1] \sum_{i=1}^n \sum_{d|\gcd(i, j)} \mu(d) \\
&= \sum_{d=1}^n [\gcd(d, k) = 1] \mu(d) \sum_{j=1}^{m/d} \sum_{i=1}^{n/d} [\gcd(j, k) = 1] \\
&= \sum_{d=1}^n [\gcd(d, k) = 1] \mu(d) \left[\frac{n}{d}\right] \sum_{j=1}^{m/d} [\gcd(j, k) = 1]
\end{aligned}$$

好啊然后发现这个东西怎么看怎么不好算， $d, k$ 长得很像但又多了个 $\mu$ 所以不能直接合并

所以开始打暴力可能需要分别计算

$$f(x) = \sum_{i=1}^x [\gcd(i, k) = 1]$$
$$g(x, k) = \sum_{i=1}^x [\gcd(i, k) = 1] \mu(i) \text{ 这里把 } k \text{ 记上是一会有用}$$

对于上面那个东西由上上道题类似的想法发现 $\gcd(k, x) \leq k$  如果 $i \perp k$  那么所有的 $i + k, i + 2k, i + 3k \dots$ 都与 $k$ 互质

所以只需预处理 $1 \sim k$ 的数中每个数与 $k$ 互质的数的个数的前缀和（特别注意这玩意不是 $\phi$ ，因为 $\phi(x)$ 表示小于等于 $x$ 的数中与 $x$ 互质的数个数而不是与 $k$ 互质的数的个数）乘上 $x/k$ ，对于 $x$ 剩下的不满 $k$ 的再单独计算即可。代码其实就一句 `int f(int x){return (x/K)*F[K]+F[x%K];}`

$f(x)$ 这么简单搞定了，那么 $g(x)$ 能否同理呢？

~~如果是这样这题就不会是最后一道子~~

很简单的例子比如 $\gcd(i, k) = 1$  那么 $\gcd(i + k, k) = 1$  但 $\mu(i + k)$ 不一定等于 $\mu(i)$

于是开始硬刚

$$\begin{aligned} g(x, k) &= \sum_{i=1}^x [\gcd(i, k) = 1] \mu(i) \\ &= \sum_{i=1}^x \mu(i) \sum_{d|\gcd(i, k)} \mu(d) \\ &= \sum_{d|k} \mu(d) \sum_{i|d} \mu(i) \\ &= \sum_{d|k} \mu(d) \sum_{p=1}^{x/d} \mu(p * d) \end{aligned}$$

然后发现当 $\gcd(p, d) \neq 1$  时 $\mu$ 为0对答案没有贡献，所以考虑 $\gcd(p, d) = 1$  时

$$\begin{aligned} g(x, k) &= \sum_{d|k} \mu^2(d) \sum_{p=1}^{x/d} \mu(p) [\gcd(p, d) = 1] \\ &= \sum_{d|k} 1 * \sum_{p=1}^{x/d} \mu(p) [\gcd(p, d) = 1] \\ &= \sum_{d|k} \sum_{p=1}^{x/d} \mu(p) [\gcd(p, d) = 1] \end{aligned}$$

突然发现这式子似曾相识。。。

$$g(x, k) = \sum_{i=1}^x \mu(i) [\gcd(i, k) = 1]$$

!!!

所以得到

$$g(x, k) = \sum_{d|k} g\left(\frac{x}{d}, d\right)$$

对于k=1时找到最开始的定义

$$g(x, 1) = \sum_{i=1}^x [\gcd(i, 1) = 1] \mu(i) \text{恒成立}$$
$$= \sum_{i=1}^x \mu(i)$$

是我们最开始讲的第一道=,然后就完了

然后又变成简单递归题了,复杂度=  $O(\text{能过})$

```
map<pii,int>MA;
inline int f(int x){return (x/K)*F[K]+F[x%K];}
inline int g(int x,int k){//比一般杜教筛多了个参数0.0
    if ((k==1 && x<MAXN) || !x) return sum[x];
    if (MA[pii(x,k)]) return MA[pii(x,k)];
    int ret=0;
    if (k==1){
        ret++;
        for (int i=2,j=2;i<=x;i=j+1){
            j=x/(x/i);
            ret-=(j-i+1)*g(x/i,1);
        }
    }else{
        for (int i=1;i*i<=k;i++) if (!(k%i)){//这里也只用枚举到sqrt(k), 因为因数是成对的
            if (mu[i])ret+=g(x/i,i); //这里防止mu[i]等于0白算一堆
            if (mu[k/i] && (i*i^k)) ret+=g(x/(k/i),k/i);//i*i!=k才再算不然贡献算两遍
        }
    }
    return MA[pii(x,k)]=ret;
}
//这里还是省略防止ctrl+c
pre();
n=min(N,M);
for (int i=1,j=1;i<=n;i=j+1){
    j=min(N/(N/i),M/(M/i));
    cur=S(j,k);
    ans+=(ll)(cur-lst)*f(M/i)*(N/i);
    lst=cur;
}
```

## 总结

杜教筛是看上去很难实际上也没那么难的东西,希望今天大家都听懂了=,