<http://www.cnblogs.com/femsub/p/5723780.html>

<http://blog.csdn.net/acdreamers/article/details/8542292>

BZOJ 2301：<http://blog.csdn.net/outer_form/article/details/50590197>

HDU 4746：

比赛的时候真是太弱了，这题没做出来，后来看了好多关于gcd(x,y)求(x,y)得对数的统计方法，才知道这个玩意该怎么做。

关于题意：定义F(n)为一个数n的素因子的个数，如12=2\*2\*3，则F(12)=3，特别的F(1)=0。

然后有10^5个询问，问满足F(gcd(i,j))<=P,(1<=x<=n,1<=y<=m)的(x,y)有多少对？

对于不会数论的人这题真的好难，对于我这种学过数论但是个不会莫比乌斯的弱渣来说更是好难。。。

假设我们枚举gcd(i,j)的值d，即gcd(i,j)=d，如果F(d)<=P，那么我们统计gcd(x,y)=d,(1<=x<=n,1<=y<=m)的(x,y)的对数，再把所有累加起来就是答案，当然了，如果我们可以O(1)得到[gcd(x,y)=d,(1<=x<=n,1<=y<=m)的(x,y)的对数]的话这个**[算法](http://lib.csdn.net/base/datastructure" \o "算法与数据结构知识库" \t "_blank)**也是可以接受的，但实际上没有O(1)的方法。。。但我们还是要考虑下这个方法

我们考虑怎样求gcd(x,y)=d，这个问题实际上等价于求gcd(x,y)=1,(1<=x<=[n/d],1<=y<=[m/d])的(x,y)的对数。

然后是怎样求gcd(x,y)=1，

我们设f(d)为gcd(x,y)=d的个数，g(d)为gcd(x,y)是d的倍数的个数，这个值很好计算：g(d)=[n/d]\*[m/d]

那么显然有g(d)=f(d)+f(2d)+f(3d)+....

然后由莫比乌斯反演，我们知道f(d)=g(d)\*u(1)+g(2\*d)\*u(2)+g(3\*d)\*u(3)+....

考虑结果ans，ans为所有f(d)且F(d)<=P的和，

我们枚举1<=i<=n，如果i是某个d的倍数且F(d)<=P，那么ans+=g(i)\*u(i/d)=[n/i]\*[m/i]\*u(i/d)。那么这个怎么计算能更快一点？

我们设G(i)为容斥因子：G(i)=sum{u(i/d) | F(d)<=P} 这个值可以nlogn预处理出来，然后我们只需要ans+=G(i)\*[n/i]\*[m/i]即可

这样的话总的复杂度为O(n\*q)还是会T的样子

然后我们注意到[n/i]\*[m/i]在一定的范围内是不变的，这个范围是[i,min(n/(n/i),m/(m/i)]，这样我们可以预处理出G(i)的前缀和，然后加快运算（复杂度网上说是sqrt(n)的。。。）

这样总的复杂度是O(q\*sqrt(n)+nlog(n))大概这样，然后就可以过了。。。呵呵呵

/\*

形式一：

g(n) = sigma(d|n, f(d))

f(n) = sigma(d|n, μ(n/d) \* g(d))

形式二：

g(n) = sigma(d|n, f(d))

f(n) = sigma(n|d, μ(n) \* g(n/d))

\*/

bool check[maxn];

int prime[maxn];

int mu[maxn];

int sum[maxn];//sum[i]； mu[]的前缀和;

int num[maxn];//num[i]: i的质因子个数;

ll cnt[maxn][20];//cnt[i][j]：表示公因子个数小于等于j时的mu的前i项和

void Moblus(){

memset(check,false,sizeof check);

mu[1]=1;

int tot=0;

for(int i=2; i<=maxn; i++){

if(!check[i]){

prime[tot++]=i;

mu[i]=-1;

num[i]=1;

}

for(int j=0; j<tot; j++){

if(i\*prime[j]>maxn)

break;

check[i\*prime[j]]=true;

num[i\*prime[j]]=num[i]+1;

if(i%prime[j]==0){

mu[i\*prime[j]]=0;

break;

}

else

mu[i\*prime[j]]=-mu[i];

}

}

for(int i=1; i<maxn; i++)

for(int j=i; j<maxn; j+=i)

cnt[j][num[i]]+=mu[j/i];

for(int i=0; i<maxn; i++)

for(int j=1; j<19; j++)

cnt[i][j]+=cnt[i][j-1];

for(int i=1; i<maxn; i++)

for(int j=0; j<19; j++)

cnt[i][j]+=cnt[i-1][j];

}

int main(){

Moblus();

int T; scanf("%d",&T);

while(T--){

int n,m,k; scanf("%d%d%d",&n,&m,&k);

k=min(k,18);

ll ans=0;

if(n>m)

swap(n,m);

for(int i=1,last=i; i<=n; i=last+1){

last=min(n/(n/i),m/(m/i));

ans+=(ll)(cnt[last][k]-cnt[i-1][k])\*(n/i)\*(m/i);

}

printf("%lld\n",ans);

}

return 0;

}