莫比乌斯反演 - ACdreamer - CSDN博客

莫比乌斯反演在数论中占有重要的地位，许多情况下能大大简化运算。那么我们先来认识莫比乌斯反演公式。

定理： 和是定义在非负整数集合上的两个函数，并且满足条件

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

，那么我们得到结论

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

在上面的公式中有一个函数，它的定义如下：

（1）若，那么

（2）若，均为互异素数，那么

（3）其它情况下

对于函数，它有如下的常见性质：

（1）对任意正整数有

$$\sum_{d|n} \mu(d)=\begin{cases} 1 & n=1 \\ 0 & n>1 \end{cases}$$

（2）对任意正整数有

$$\sum_{d|n} \frac{\mu(d)}{d}=\frac{\varphi(n)}{n}$$

线性筛选求莫比乌斯反演函数代码。

[cpp]view plaincopy

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

有了上面的知识，现在我们来证明莫比乌斯反演定理。

证明

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

证明完毕！

嗯，有了莫比乌斯反演，很多问题都可以简化了，接下来我们来看看莫比乌斯反演在数论中如何简化运算的。

题目： http://bz.cdqzoi.com/JudgeOnline/problem.php?id=2818

题意：给一个正整数，其中，求使得为质数的的个数，。

分析：对于本题，因为是使得为质数，所以必然要枚举小于等于的质数，那么对于每一个质数，只需要求在区间中，满足有序对互质的对数。

也就是说，现在问题转化为：在区间中，存在多少个有序对使得互质，这个问题就简单啦，因为是有序对，不妨设，那么我们如果枚举每一个，小于有多少个与互素，这正是欧拉函数。所以我们可以递推法求欧拉函数，将得到的答案乘以2即可，但是这里乘以2后还有漏计算了的，那么有哪些呢？

是且为素数的情况，再加上就行了。

代码：

[cpp]view plaincopy

```cpp
#include
#include
#include
#include
usingnamespace std;
typedeflonglong LL;
constint N = 10000010;
bitset prime;
LL phi[N];
LL f[N];
int p[N];
int k;
void isprime()
{
    k = 0;
    prime.set();
for(int i=2; i
    {
if(prime[i])
        {
            p[k++] = i;
for(int j=i+i; j
                prime[j] = false;
        }
    }
}
void Init()
{
for(int i=1; i
for(int i=2; i>= 1;
for(int i=3; i
    {
if(phi[i] == i)
        {
for(int j=i; j
                phi[j] = phi[j] - phi[j] / i;
        }
    }
    f[1] = 0;
for(int i=2;i
        f[i] = f[i-1] + (phi[i]<<1);
}
LL Solve(int n)
{
    LL ans = 0;
for(int i=0; i<=n; i++)
        ans += 1 + f[n/p[i]];
return ans;
}
int main()
{
    Init();
    isprime();
```

```
int n;
    scanf("%d",&n);
    printf("%I64d\n",Solve(n));
return 0;
}
```

嗯，上题不算太难，普通的欧拉函数就可以搞定，接下来我们来看看它的升级版。

题意：给定两个数和，其中，，求为质数的有多少对？其中和的范围是。

分析：本题与上题不同的是和不一定相同。在这里我们用莫比乌斯反演来解决，文章开头也说了它能大大简化运算。我们知道莫比乌斯反演的一般描述为：

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

其实它还有另一种描述，本题也是用到这种。那就是：

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

好了，到了这里，我们开始进入正题。。。

对于本题，我们设

为满足且和的的对数

为满足且和的的对数

那么，很显然，反演后得到

$$f(x)=\sum_{x|d}\mu(\frac{d}{x})F(d)=\sum_{x|d}\mu(\frac{d}{x})\frac{n}{d}\bullet\frac{m}{d}$$

因为题目要求是为质数，那么我们枚举每一个质数，然后得到

$$ans=\sum_{p}^{min(n,m)}\left(\sum_{d}^{min(n,m)}\mu(d)\frac{n}{pd}\bullet\frac{m}{pd}\right)$$

如果直接这样做肯定TLE，那么我们必须优化。

我们设，那么继续得到

$$ans=\sum_{T=1}^{min(m,n)}\frac{n}{T}\bullet\frac{m}{T}\left(\sum_{p|T}\mu(\frac{T}{p})\right)$$

。

到了这里，可以看出如果我们可以先预处理出所有的对应的的值，那么本题就解决了。

我们设

$$sum(x)=\sum_{p|x}\mu(\frac{x}{p})$$

，注意这里为素数，。

那么，我们枚举每一个，得到

$$g(kx)=\mu(\frac{kx}{p})$$

，现在分情况讨论：

（1）如果整除，那么得到

$$g(kx)=\begin{cases}\mu(x) & k=p\\ 0 & k!=p\end{cases}$$

（2）如果不整除，那么得到

$$g(kx)=\begin{cases}-\mu(x) & k=p\\ \mu(x)-g(x) & k!=p\end{cases}$$

[cpp]view plaincopy

```
#include
#include
#include
usingnamespace std;
typedeflonglong LL;
constint N = 10000005;
```

```
bool vis[N];
int p[N];
int cnt;
int g[N],u[N],sum[N];
void Init()
{
    memset(vis,0,sizeof(vis));
    u[1] = 1;
    cnt = 0;
for(int i=2;i
    {
if(!vis[i])
        {
            p[cnt++] = i;
            u[i] = -1;
            g[i] = 1;
        }
for(int j=0;j
        {
            vis[i*p[j]] = 1;
if(i%p[j])
            {
                u[i*p[j]] = -u[i];
                g[i*p[j]] = u[i] - g[i];
            }
else
            {
                u[i*p[j]] = 0;
                g[i*p[j]] = u[i];
break;
            }
        }
    }
    sum[0] = 0;
for(int i=1;i
        sum[i] = sum[i-1] + g[i];
}
int main()
{
    Init();
int T;
    scanf("%d",&T);
while(T--)
    {
        LL n,m;
        cin>>n>>m;
if(n > m) swap(n,m);
        LL ans = 0;
for(int i=1,last;i<=n;i=last+1)
        {
            last = min(n/(n/i),m/(m/i));
            ans += (n/i)*(m/i)*(sum[last]-sum[i-1]);
        }
        cout<<
    }
return 0;
}
```