原创 莫比乌斯反演入门讲解

莫比乌斯反演实际上是一两个公式定理的运用，自认为想要掌握它的话，其中的证明还是有必要了解的。看过网上一些博客，感觉都只证明了一半，没
个定理完全证明出来。然而我最近在正好在学习初等数论，发现完全证明这个定理实际上并不需要很多知识，特此填坑。另外加上一些应用以构成完整

实际上，这个定理的证明用到了一点数论函数相关知识。前置技能在此

https://blog.csdn.net/tomandjake_/article/details/81083051

内容并不多，自认为把这一页笔记内容学会，自己就可以把莫比乌斯反演的证明当练习题一样独立完成。接下来我还是会解释，当然，如果觉得我的解
可以自己看这个笔记学习，我甚至都推荐自己看笔记，因为如果熟悉这样的数学语言，其实看的很快而且会有自己的认识。而且其中给出了欧拉函数的
过欧拉函数的话，看这个也可以更加高效地学完这一块内容。(注意笔记中(m,n)=gcd(m,n))

好，我们进入正题。首先引入三个定义

Definition 1 可乘函数：算术函数f，满足 只要gcd(m,n) =1,就有f(mn)=f(m)f(n)。

没错, 比其他可乘函数宽泛一点，只要在gcd(m,n)=1条件下满足可乘就行。然后我们知道由唯一分解定理，任何正整数n可分解为若干素数的幂方积 $n=p_1^{\alpha 1}p_2^{\alpha 2}\ldots\ldots p_k^{\alpha k}$,所以若$\theta$为可乘函数，有

$$\theta(n)=\theta\left(p1^{\alpha 1}\right)\theta\left(p2^{\alpha 2}\right)\ldots\theta\left(pk^{\alpha k}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Definition 2 $\sum_{d|n}$： 代表对n的所有正因子求和，如

$$\sum_{d|4}d^2=1^2+2^2+4^2$$

Definition 3 莫比乌斯函数：

$$\mu(1)=1,\ \ \mu(n)=\mu\left(p1^{\alpha 1}\right)\mu\left(p2^{\alpha 2}\right)\ldots\mu\left(pk^{\alpha k}\right)=\begin{cases}(-1)^k & 若\alpha 1=\alpha 2=..=\alpha k=1\\ 0 & otherwise\end{cases}$$

定义有点奇怪，关于这个函数，之后我们可以看到它的作用, 必须提到的是，可以证明，**莫比乌斯函数是可乘函数**。

然后是我们的目标----莫比乌斯反演定理：

Theorem 1 莫比乌斯反演定理: F(n)和f(n)为算术函数，若他们满足

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

则有

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

证明如下：

首先给出一个定理，

Theorem 2

$\psi(n)=\sum_{d|n}\theta(d)$ 为可乘函数, 由(1)式及可乘函数定义可知

$$\Longrightarrow \psi(n)\ =\left(1+\theta(p1)+\theta\left(p1^2\right)+\ldots\theta\left(p1^{\alpha 1}\right)\right)\left(1+\theta(p2)+\theta\left(p2^2\right)+..+\theta\left(p1^{\alpha 2}\right)\right)\ldots$$

至于 $\sum_{d|n}\theta(d)$ 为可乘函数的证明，可参见我的笔记。

之后可以得到下一个定理

Theorem 3 若θ可乘，$\mu\theta$可乘，$\sum_{d|n}\mu(d)\theta(d)$也可乘，从而

$$\sum_{d|n}\mu(d)\theta(d)=(1-\theta(p1))(1-\theta(p2))\ldots.(1-\theta(pk))$$

特别有：

$$\theta\equiv 1,\sum_{d|n}\mu(d)=\begin{cases}1 & n=1\\ 0 & otherwise\end{cases}\qquad (3)$$

$$\theta=\frac{1}{d},\sum_{d|n}\mu(d)=\left(1-\frac{1}{p1}\right)\left(1-\frac{1}{p2}\right)\ldots.\qquad (4)$$

定理3证明：

$\theta(n)$和$\mu(n)$为可乘函数 $\Longrightarrow$ $\mu(n)\theta(n)$为可乘函数

由定理2得，$\sum_{d|n}\mu(d)\theta(d)$也为可乘函数，且

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

$$\left(1+\mu(p_1)\theta(p_1)+\mu\left(p_1^2\right)\theta\left(p_1^2\right)+\ldots+\mu\left(p_1^{\alpha1}\right)\theta\left(p_1^{\alpha1}\right)\right)\left((1+\mu(p_2)\theta(p_2))+\ldots+\left(1+\mu\left(p_2^{\alpha2}\right)\theta\left(p_2^{\alpha2}\right)\right)\right)\ldots$$

而由莫比乌斯函数定义，当$\alpha>1$时，$\mu\left(p^{\alpha}\right)=0$，由此每个大括号中只需保留前两项，并且当$\alpha=1$时$\mu\left(p^{\alpha}\right)=\mu(p)=-1$，

$$\Longrightarrow\sum_{d|n}\mu(d)\theta(d)=(1-\theta(p1))(1-\theta(p2))\ldots.(1-\theta(pk))$$

于是就有

$$\theta(d)\equiv 1,\sum_{d|n}\mu(d)=\begin{cases}1 & n=1\\ 0 & otherwise\end{cases}\qquad (3)$$

$$\theta(d)=\frac{1}{d},\sum_{d|n}\mu(d)=\left(1-\frac{1}{p1}\right)\left(1-\frac{1}{p2}\right)\ldots.\qquad (4)$$

最后我们直接证明莫比乌斯反演（几个定理定义学习完，这其实就算是个练习题啦~）：

$$\sum_{d|n}\mu(d)F\left(\frac{n}{d}\right)=\sum_{d|n}\mu(d)\sum_{d1|\frac{n}{d}}f(d1)\text{，由}\left(\begin{cases}d|n\\ d1|\frac{n}{d}\end{cases}\Leftrightarrow\begin{cases}d|\frac{n}{d1}\\ d1|n\end{cases}\right)\Longrightarrow$$

$$\sum_{d|n}\mu(d)\sum_{d1|\frac{n}{d}}f(d1)=\sum_{d|\frac{n}{d1}}\mu(d)\sum_{d1|n}f(d1)=\sum_{d1|n}f(d1)\sum_{d|\frac{n}{d1}}\mu(d)=f(n)\quad\text{（注意定理三中的（3）式）}$$

可以看到，莫比乌斯反演提供了一个F(n)与f(n)的连接，而他们之间的桥梁就是莫比乌斯函数μ。之后我们会看到具体的例子。这里先给出线性筛莫比乌
码：

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

应用举例：POJ 3904

题目给出n和n个正整数，问能找出多少个不同的四元组(a,b,c,d)使得该四元组的最大公因数为1.

分析：

首先要提的是，我们做题目往往用的是莫比乌斯反演的另外一种形式：

若有 $F(n)=\sum_{n|d} f(d)$ 则有 $f(n)=\sum_{n|d} \mu\left(\frac{d}{n}\right)F(d)$

既然要用到莫比乌斯反演，我们首先就要找到合适的F和f。实际上，F和f的关系在于整除。我们可以假设

F(n)为有多少个四元组满足gcd(a,b,c,d)=n的整数倍

f(n)为有多少个四元组满足gcd(a,b,c,d)=n

所以我们的目标就是求f(1), 而实际上可以看出F与f构成了一组莫比乌斯变换对。有了反演公式，我们可以通过求F来求f，而这里面F很好求，要求F（n） 原数组中数出到能被n整除的数的个数m, 则C（m,4）就是F（n）

而由公式，由此我们可以发现 $f(1)=\mu(1)F(1)+\mu(2)F(2)+......+\mu(maxn)F(maxn)$

直接计算即可。

代码：

```
#include<iostream>
#include<cstdio>
#include<string>
#include<cstring>
#include<vector>
#include<stack>
#include<algorithm>
#include<map>
#include<set>
#include<queue>
#include<sstream>
#include<cmath>
#include<iterator>
#include<bitset>
#include<stdio.h>
using namespace std;
#define _for(i,a,b) for(int i=(a);i<(b);++i)
#define _rep(i,a,b) for(int i=(a);i<=(b);++i)
typedef long long LL;
const int INF = 1 << 30;
const int MOD = 1e9 + 7;
const int maxn = 10005;

int n, a[maxn], tot[maxn];
int mu[maxn], vis[maxn];
int primes[maxn], cnt;
void get_mu() {
    memset(vis, 0, sizeof(vis));
    memset(mu, 0, sizeof(mu));
    cnt = 0; mu[1] = 1;
    for (int i = 2; i <= maxn; ++i) {
        if (!vis[i]) { primes[cnt++] = i; mu[i] = -1; }
        for (int j = 0; j<cnt&&primes[j] * i <= maxn; ++j) {
            vis[primes[j] * i] = 1;
            if (i%primes[j] == 0)break;
            mu[i*primes[j]] = -mu[i];
        }
    }
}

void get_tot() {
    memset(tot, 0, sizeof(tot));
    for (int i = 0; i<n; ++i) {
```

```
            int x = a[i];
            int m = sqrt(x);
            for (int j = 1; j <= m; ++j) {
                if (x%j == 0)tot[j]++, tot[x / j]++;
            }
            if (m*m == x)tot[m]--;
        }
}
LL Cn4(int m) {
        if (m == 0)return 0;
        return 1ll * m*(m - 1)*(m - 2)*(m - 3) / 24;
}
int main()
{
        //freopen("C:\\Users\\admin\\Desktop\\in.txt", "r", stdin);
        //freopen("C:\\Users\\admin\\Desktop\\out.txt", "w", stdout);
        get_mu();
        while (~scanf("%d", &n)) {
            for (int i = 0; i<n; ++i) scanf("%d", &a[i]);
            get_tot();
            LL ans = 0;
            for (int i = 1; i<maxn; ++i) {
                ans += 1ll * mu[i] * Cn4(tot[i]);
            }
            printf("%I64d\n", ans);
        }

        return 0;
}
```

这样莫比乌斯反演就算是入门了吧。