Project for Discrete Mathematics

Name: 张嘉浩 SID: 12010423

Introduction and Abstract

Carl Friedrich Gauss, one of the greatest mathematicians had claimed: "Mathematics is the queen of the sciences and number theory is the queen of mathematics." Fermat's Little Theorem plays a crucial part in number theory. Having learnt it from Discrete Mathematics, its elegance in solving problems such as calculating $a^n \pmod p$ for a prime p, especially when n is sufficiently large is highly appreciated. However, we need to turn to other approaches when the modulo is not a prime. Here we introduce the Euler's Theorem, also known as the extended Fermat's Little Theorem, which needs to first specify the φ function, a representative of the arithmetic function. Among all the arithmetic functions, there are some functions that own a special and important property——multiplicativity. For two multiplicative functions, one defines a new multiplicative function $f * g$, namely the Dirichlet Convolution. Thus, Möbius Inversion can be then derived, which plays an important role in Analytic Number Theory and other algorithm problems. At the end of the project, a specific application using Möbius Inversion which is related to gcd is illustrated.

Fermat's Little Theorem

Let a be a positive integer, p be a prime and $\gcd(a, p) = 1$. Then $a^{p-1} \equiv 1 (\bmod p)$.

Proof of Fermat's Little Theorem

We use Mathematical Induction to complete the proof.

1. Base case: $0^p \equiv 0 \pmod p$ is trivial.
2. Inductive case: we must show that if the theorem is true for $a = k$, then it is also true for $a = k + 1$.

   We utilize the lemma that: For any integers x and y and for any prime p, $(x + y)^p \equiv x^p + y^p (\bmod p)$.

   Proof of Lemma:

   The binomial coefficients are all integers. The numerator contains a factor p by the definition of factorial. When $0 < i < p$, neither of the terms in the denominator includes a factor of p (relying on the primality of p), leaving the coefficient itself to possess a prime factor of p from the numerator, implying that

$$\binom{p}{i} \equiv 0 \pmod p, \qquad 0 < i < p.$$

Modulo p, this eliminates all but the first and last terms of the sum on the right-hand side of the binomial theorem for prime p.

Assume $k^p \equiv k(\mod p)$, and consider $(k+1)^p$. By the lemma we have

$$(k+1)^p \equiv k^p + 1^p \pmod p.$$

Using the induction hypothesis, we have that$k^p \equiv k(\mod p)$ and trivially $1^p = 1$, thus

$$(k+1)^p \equiv k+1 \pmod p,$$

which is the statement of the theorem for a = k+1. Hence we finished the proof of induction.

Now we show that if a be a positive integer, p be a prime and $\gcd(a,p) = 1$. Then $a^{p-1} \equiv 1(\text{mod } p)$ from induction.

From the form of Fermat's Little Theorem, we would like to extend p, which leads to Euler's Theorem. Actually, Euler reaches the Theorem by the process of proving Fermat's Little Theorem. But before moving to Euler's Theorem, we need to introduce φ function first.

φ Function

φ Function is also known as Euler's Function.

$\varphi(n)$ is defined to be the number of positive integers not greater than n that is relatively prime to n, which can be calculated as

$$\varphi(n) = n\prod_{p|n}(1 - \frac{1}{p})$$

Properties:

The following properties of Euler totient function are sufficient to calculate it for any number:

- If p is a prime number, then $gcd(p,q) = 1$ for all $1 \leq q < p$. Therefore we have:

$$\varphi(p) = p - 1$$

- If p is a prime number and $k \geq 1$, then there are exactly p^k/p numbers between 1 and p^k that are divisible by p. Which gives us:

$$\varphi(p^k) = p^k - p^{k-1}$$

- If a and b are relatively prime, then:

$$\varphi(ab) = \varphi(a) \times \varphi(b)$$

Multiplicative

Definition of multiplicativity

An arithmetic function f is called multiplicative if $f(mn) = f(m)f(n)$ whenever m, n are relatively prime.

Euler’s phi function φ is multiplicative. In other words, if $gcd(m,n) = 1$ then $\varphi(mn) = \varphi(m)\varphi(n)$

Proof of the multiplicativity of φ

We make a rectangular table of the numbers 1 to mn with m rows and n columns, as follows:

1	**m+1**	**2m+1**	**...**	$(n-1)m+1$
2	m+2	2m+2	...	$(n-1)m+2$
3	m+3	2m+3	...	$(n-1)m+3$
...	...	...	...	...
m	2m	3m	...	mn

The numbers in the rth row of this table are of the form $km + r$ as k runs from 0 to $m - 1$.

Let $d = \gcd(r, m)$. If $d > 1$ then no number in the rth row of the table is relatively prime to mn, since $d|(km + r)$ for all k. So to count the residues relatively prime to mn we need only to look at the rows indexed by values of r such that $gcd(r, m)$ = 1, and there are $\varphi(m)$ such rows.
If $gcd(r, m) = 1$ then every entry in the rth row is relatively prime to m, since $gcd(km + r, m) = 1$ by the Euclidean algorithm. Thus, exactly $\varphi(n)$ of them will be relatively prime to n, and thus relatively prime to mn.
We have shown that there are $\varphi(m)$ rows in the table which contain numbers relatively prime to mn, and each of those contain exactly $\varphi(n)$ such numbers.
So there are, in total, $\varphi(m)\varphi(n)$ numbers in the table which are relatively prime to mn. This proves the theorem.

Euler's Theorem

Let a , m be positive integers and $\gcd(a, m) = 1$. Then $a^{\varphi(m)} \equiv 1(\bmod m)$, where φ is Euler's function.

Proof of Euler's Theorem

Let $R = x_1, x_2, \ldots, x_{\varphi(n)}$ be a reduced residue system (mod n) and let a be any integer coprime to n. The proof hinges on the fundamental fact that multiplication by a permutes the x_i: in other words if $ax_j \equiv ax_k(\mod n)$then $j = k$. That is, the sets R and $aR = ax_1, ax_2, \ldots, ax_{\varphi(n)}$, considered as sets of congruence classes (mod n), are identical (as sets—they may be listed in different orders), so the product of all the numbers in R is congruent (mod n) to the product of all the numbers in aR:

$$\prod_{i=1}^{\varphi(n)} x_i \equiv \prod_{i=1}^{\varphi(n)} ax_i = a^{\varphi(n)} \prod_{i=1}^{\varphi(n)} x_i \pmod n,$$

and using the cancellation law to cancel each x_i gives Euler's theorem.

Q.E.D

Now that we have the φ function, we start to consider all the so-called "number theory functions", more formally known as "arithmetic functions".

Arithmetic Functions

An arithmetic function is a function $f : \mathbb{N} \to \mathbb{C}$ that maps $\mathbb{N}$ to a subset of $\mathbb{C}$ (usually $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$).

They describe arithmetic properties of numbers and are widely used in the field of number theory. Arithmetic functions are different from typical functions in that they cannot usually be described by simple formulas, so they are often evaluated in terms of their average or asymptotic behavior.

Among all the arithmetic functions, there are some functions which own a special and important property.

Multiplicative Functions

Definition

In number theory, a **multiplicative** function is an arithmetic function $f(n)$ of a positive integer n with the property that $f(1) = 1$ and $f(ab) = f(a)f(b)$ whenever a and b are coprime.
Moreover, an arithmetic function $f(n)$ is said to be **completely multiplicative** if $f(1) = 1$ and $f(ab) = f(a)f(b)$ holds for all positive integers a and b, even when they are not coprime.

Example: σ(n)

In number theory, a divisor function is an arithmetic function related to the divisors of an integer. The sum of positive divisors function $\sigma z(n)$, for a real or complex number z, is defined as the sum of the $z\text{th}$ powers of the positive divisors of n. It can be expressed in sigma notation as $\sigma_z(n) = \sum_{d|n} d^z$.

From the formulas above, we can easily see that σ(n) is a multiplicative function.

The addition of arithmetic functions is trivial: $(f + g)(n) := f(n) + g(n)$. On the other hand, the "multiplication" of two arithmetic function is much more difficult and attractive.

Dirichlet Convolution

Suppose functions f, g are arithmetic functions. Denote the Dirichlet convolution of f and g as $f * g$, and define it as follows:

$$(f * g)(n) := \sum_{d|n} f(d)g(\frac{n}{d}).$$

Properties

Dirichlet Convolution satisfies the following properties:

1. Associative

$$(f * g) * h = f * (g * h)$$

2. Distributive

$$f * (g + h) = f * g + f * h$$

3. Commutative

$$f * g = g * f$$

4. Identity element

$$f * \varepsilon = \varepsilon * f = f$$

5. The convolution of two multiplicative functions is still multiplicative.

Dirichlet Inverse

Given an arithmetic function f its Dirichlet inverse $g = f^{-1}$ may be calculated recursively: the value of $g(n)$ is in terms of $g(m)$ for $m < n$.
For $n = 1$:
$(f * g)(1) = f(1)g(1) = \varepsilon(1) = 1$, so $g(1) = 1/f(1)$. This implies that f does not have a Dirichlet inverse if $f(1) = 0$.

For $n = 2$:
$(f * g)(2) = f(1)g(2) + f(2)g(1) = \varepsilon(2) = 0$
$g(2) = -(f(2)g(1))/f(1)$

For $n = 3$:
$(f * g)(3) = f(1)g(3) + f(3)g(1) = \varepsilon(3) = 0$
$g(3) = -(f(3)g(1))/f(1)$

For $n = 4$:
$(f * g(4) = f(1)g(4) + f(2)g(2) + f(4)g(1) = \varepsilon(4) = 0$
$g(4) = -(f(4)g(1) + f(2)g(2))/f(1)$

In general, for $n > 1$:

$$g(n) \ = \ \frac{-1}{f(1)} \sum_{\substack{d \mid n \\ d < n}} f\left(\frac{n}{d}\right) g(d).$$

Properties

The following properties of the Dirichlet inverse hold:

1. The function f has a Dirichlet inverse if and only if $f(1) \neq 0$.
2. The Dirichlet inverse of a multiplicative function is again multiplicative.
3. The Dirichlet inverse of a Dirichlet convolution is the convolution of the inverses of each function: $(f * g)^{-1} = f^{-1} * g^{-1}$.
4. A multiplicative function f is completely multiplicative if and only if $f^{-1}(n) = \mu(n)f(n)$.
5. If f is completely multiplicative then $(f \cdot g)^{-1} = f \cdot g^{-1}$whenever $g(1) \neq 0$ and where $\cdot$ denotes pointwise multiplication of functions.

By using Dirichlet Inverse, we can derive related function and formula about Möbius Inversion, which is extremely important in Analytic Number Theory.

Möbius function

We illustrated Möbius function before we come to the Möbius Inversion.
For any positive integer n, define $\mu(n)$ as the sum of the primitive nth roots of unity. It has values in $\{-1,0,1\}$ depending on the factorization of n into prime factors:

- $\mu(n)=+1$ if n is a square-free positive integer with an even number of prime factors.
- $\mu(n)=-1$ if n is a square-free positive integer with an odd number of prime factors.
- $\mu(n)=0$ if n has a squared prime factor.

The Möbius function can alternatively be represented as

$$\mu(n)=\delta_{\omega(n)\Omega(n)}\lambda(n),$$

where δ is the Kronecker delta, $\lambda(n)$ is the Liouville function, $\omega(n)$ is the number of distinct prime divisors of n, and $\Omega(n)$ is the number of prime factors of n, counted with multiplicity.

Möbius Inversion

Suppose functions f, g are arithmetic functions and that

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

for every integer $n>0$

Then we have

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

for every integer $n>0$

In the language of Dirichlet convolutions, the first formula may be written as $g=f*1$.
where $*$ denotes the Dirichlet convolution, and 1 is the constant function $1(n)=1$. The second formula is then written as $f=\mu*g$..

The theorem follows because $*$ is (commutative and) associative, and $1*\mu=\varepsilon$, where ε is the identity function for the Dirichlet convolution, taking values $\varepsilon(1)=1, \varepsilon(n)=0$ for all $n>1$. Thus

$\mu*g=\mu*(1*f)=(\mu*1)*f=\varepsilon*f=f.$

There is also a product version of the summation-based Möbius inversion formula stated above:
$g(n)=\prod_{d|n}f(d) \iff f(n)=\prod_{d|n}g\left(\frac{n}{d}\right)^{\mu(d)}, \forall n\geq 1.$

Applications of Möbius Inversion in computer science

Example

Given an integer N. Calculate the number of pairs (x, y) such that $1 \le x, y \le N$ $(N \le 1e7)$ and $gcd(x, y)$ is a prime.

Solution 1: Euler's function

We can list the prime p, calculate φ of n/p, then multiply by 2. Notice that we should deal with cautious when $x = y$ & x, y are prime.

```
#include<bits/stdc++.h>
#define ll long long
using namespace std;
int n,cnt;
ll phi[10000001];
int vis[10000001];
int prime[10000001];
ll sum=0;
void init(){
    for(register int i=2;i<=n;i++){
        if(!vis[i]){
            prime[++cnt]=i;
            phi[i]=i-1;
        }
        for(register int j=1;j<=cnt&&i*prime[j]<=n;j++){
            int k=i*prime[j];
            vis[k]=1;
            if(i%prime[j]==0){
                phi[k]=phi[i]*prime[j];
                break;
            }
            else{
                phi[k]=phi[i]*(prime[j]-1);
            }
        }
        phi[i]=phi[i-1]+(phi[i]<<1);
    }
}
int main(){
    scanf("%d",&n);
    init();
    for(register int i=1;i<=cnt;i++){
        sum+=phi[n/prime[i]]+1;
    }
    printf("%lld",sum);
    return 0;
}
```

Solution 2:
The solution 1 using φ function seems satisfying. Nevertheless, when the prerequisite comes to $1 \le x \le n, 1 \le y \le m$, Euler function encountered a stumbling block. Now is when Möbius Inversion reveal its essence.
Suppose we have $f(n) = \sum_{d|n}[gcd(x, y) == d]$, let $F(n) = \sum_{d|n} f(d)$. By the Möbius Inversion, we have $f(n) = \sum_{d|n} \mu(n/d) * F(d)$, where $F(d) = (n/d) * (n/d)$ by its definition.
Thus

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

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<cstdlib>
#include<cmath>
#include<vector>
#include<queue>
#define ll long long
using namespace std;
int mu[10000001];
int cnt;
bool vis[10000001];
int prime[5000001];
void init(int n){
    mu[1]=1;
    for(int i=2;i<=n;i++){
        if(!vis[i]){
            prime[++cnt]=i;
            mu[i]=-1;
        }
        for(int j=1;j<=cnt&&i*prime[j]<=n;j++){
            int k=i*prime[j];
            vis[k]=1;
            if(i%prime[j]){
                mu[k]=-mu[i];
            }
            else{
                mu[k]=0;
                break;
            }
        }
    }
}
int main(){
    ll n;
    scanf("%lld",&n);
    init(n);
    ll ans=0;
    for(ll i=1;i<=cnt;i++){
        ll lim=n/prime[i];
        for(ll j=1;j<=lim;j++){
            ans+=mu[j]*((lim)/j)*((lim)/j);
        }
    }
    printf("%lld",ans);
    return 0;
}
```

Summary and Improvement

I started this project from Fermat's Little Theorem, which is taught from textbook and course assignments. By delving it deeper and deeper, I successively learnt more about Euler's Theorem, arithmetic functions, Dirichlet Convolution and eventually Möbius Inversion. I've learnt so much by diving into it and searching for relevant materials and refer to textbook about the number theory. Although other topics such as dijkstra and other algorithm seems easier for me to accomplish, since I've completed some of them in algorithm courses, I decided to dive into a field that I had little background, "the queen of Mathematics" —— number theory. I started to appreciate its charm in Discrete Mathematics courses, dived into it for weeks, and eventually completed this project. Nevertheless, due to the lack of background in number theory, I've only accomplished some basic illustrations, proofs and applications. Hopefully after learning more about number theory, I'll keep improving this project during this summer vacation.

References

1. Euler’s totient function - Algorithms for Competitive Programming. (2020). Cp-Algorithms. https://cp-algorithms.com/algebra/phi-function.html
2. R. (2018). Discrete Mathematics and Its Applications (Eighth Edition). Mc Graw Hill Education (Uk).
3. Spector, D. (1989). Multiplicative functions, Dirichlet convolution, and quantum systems. Physics Letters A, 140(6), 311–316. https://doi.org/10.1016/0375-9601(89)90626-9
4. Wikipedia contributors. (2021, October 19). Möbius inversion formula. Wikipedia. https://en.wikipedia.org/wiki/M%C3%B6bius_inversion_formula
5. Wikipedia contributors. (2022, March 21). Dirichlet convolution. Wikipedia. https://en.wikipedia.org/wiki/Dirichlet_convolution-