

Chapter 21

Number theory

Number theory is a branch of mathematics that studies integers. Number theory is a fascinating field, because many questions involving integers are very difficult to solve even if they seem simple at first glance.

As an example, consider the following equation:

$$x^3 + y^3 + z^3 = 33$$

It is easy to find three real numbers x , y and z that satisfy the equation. For example, we can choose

$$\begin{aligned}x &= 3, \\y &= \sqrt[3]{3}, \\z &= \sqrt[3]{3}.\end{aligned}$$

However, it is an open problem in number theory if there are any three *integers* x , y and z that would satisfy the equation [6].

In this chapter, we will focus on basic concepts and algorithms in number theory. Throughout the chapter, we will assume that all numbers are integers, if not otherwise stated.

Primes and factors

A number a is called a **factor** or a **divisor** of a number b if a divides b . If a is a factor of b , we write $a \mid b$, and otherwise we write $a \nmid b$. For example, the factors of 24 are 1, 2, 3, 4, 6, 8, 12 and 24.

A number $n > 1$ is a **prime** if its only positive factors are 1 and n . For example, 7, 19 and 41 are primes, but 35 is not a prime, because $5 \cdot 7 = 35$. For every number $n > 1$, there is a unique **prime factorization**

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k},$$

where p_1, p_2, \dots, p_k are distinct primes and $\alpha_1, \alpha_2, \dots, \alpha_k$ are positive numbers. For example, the prime factorization for 84 is

$$84 = 2^2 \cdot 3^1 \cdot 7^1.$$

The **number of factors** of a number n is

$$\tau(n) = \prod_{i=1}^k (\alpha_i + 1),$$

because for each prime p_i , there are $\alpha_i + 1$ ways to choose how many times it appears in the factor. For example, the number of factors of 84 is $\tau(84) = 3 \cdot 2 \cdot 2 = 12$. The factors are 1, 2, 3, 4, 6, 7, 12, 14, 21, 28, 42 and 84.

The **sum of factors** of n is

$$\sigma(n) = \prod_{i=1}^k (1 + p_i + \dots + p_i^{\alpha_i}) = \prod_{i=1}^k \frac{p_i^{\alpha_i+1} - 1}{p_i - 1},$$

where the latter formula is based on the geometric progression formula. For example, the sum of factors of 84 is

$$\sigma(84) = \frac{2^3 - 1}{2 - 1} \cdot \frac{3^2 - 1}{3 - 1} \cdot \frac{7^2 - 1}{7 - 1} = 7 \cdot 4 \cdot 8 = 224.$$

The **product of factors** of n is

$$\mu(n) = n^{\tau(n)/2},$$

because we can form $\tau(n)/2$ pairs from the factors, each with product n . For example, the factors of 84 produce the pairs $1 \cdot 84$, $2 \cdot 42$, $3 \cdot 28$, etc., and the product of the factors is $\mu(84) = 84^6 = 351298031616$.

A number n is called a **perfect number** if $n = \sigma(n) - n$, i.e., n equals the sum of its factors between 1 and $n - 1$. For example, 28 is a perfect number, because $28 = 1 + 2 + 4 + 7 + 14$.

Number of primes

It is easy to show that there is an infinite number of primes. If the number of primes would be finite, we could construct a set $P = \{p_1, p_2, \dots, p_n\}$ that would contain all the primes. For example, $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, and so on. However, using P , we could form a new prime

$$p_1 p_2 \cdots p_n + 1$$

that is larger than all elements in P . This is a contradiction, and the number of primes has to be infinite.

Density of primes

The density of primes means how often there are primes among the numbers. Let $\pi(n)$ denote the number of primes between 1 and n . For example, $\pi(10) = 4$, because there are 4 primes between 1 and 10: 2, 3, 5 and 7.

It is possible to show that

$$\pi(n) \approx \frac{n}{\ln n},$$

which means that primes are quite frequent. For example, the number of primes between 1 and 10^6 is $\pi(10^6) = 78498$, and $10^6 / \ln 10^6 \approx 72382$.

Conjectures

There are many *conjectures* involving primes. Most people think that the conjectures are true, but nobody has been able to prove them. For example, the following conjectures are famous:

- **Goldbach's conjecture:** Each even integer $n > 2$ can be represented as a sum $n = a + b$ so that both a and b are primes.
- **Twin prime conjecture:** There is an infinite number of pairs of the form $\{p, p + 2\}$, where both p and $p + 2$ are primes.
- **Legendre's conjecture:** There is always a prime between numbers n^2 and $(n + 1)^2$, where n is any positive integer.

Basic algorithms

If a number n is not prime, it can be represented as a product $a \cdot b$, where $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$, so it certainly has a factor between 2 and $\lfloor \sqrt{n} \rfloor$. Using this observation, we can both test if a number is prime and find the prime factorization of a number in $O(\sqrt{n})$ time.

The following function `prime` checks if the given number n is prime. The function attempts to divide n by all numbers between 2 and $\lfloor \sqrt{n} \rfloor$, and if none of them divides n , then n is prime.

```
bool prime(int n) {
    if (n < 2) return false;
    for (int x = 2; x*x <= n; x++) {
        if (n%x == 0) return false;
    }
    return true;
}
```

The following function `factors` constructs a vector that contains the prime factorization of n . The function divides n by its prime factors, and adds them to the vector. The process ends when the remaining number n has no factors between 2 and $\lfloor \sqrt{n} \rfloor$. If $n > 1$, it is prime and the last factor.

```
vector<int> factors(int n) {
    vector<int> f;
    for (int x = 2; x*x <= n; x++) {
        while (n%x == 0) {
            f.push_back(x);
            n /= x;
        }
    }
    if (n > 1) f.push_back(n);
    return f;
}
```

Note that each prime factor appears in the vector as many times as it divides the number. For example, $24 = 2^3 \cdot 3$, so the result of the function is $[2, 2, 2, 3]$.

Sieve of Eratosthenes

The **sieve of Eratosthenes** is a preprocessing algorithm that builds an array using which we can efficiently check if a given number between $2 \dots n$ is prime and, if it is not, find one prime factor of the number.

The algorithm builds an array `sieve` whose positions $2, 3, \dots, n$ are used. The value `sieve[k] = 0` means that k is prime, and the value `sieve[k] ≠ 0` means that k is not a prime and one of its prime factors is `sieve[k]`.

The algorithm iterates through the numbers $2 \dots n$ one by one. Always when a new prime x is found, the algorithm records that the multiples of x ($2x, 3x, 4x, \dots$) are not primes, because the number x divides them.

For example, if $n = 20$, the array is as follows:

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	2	0	3	0	2	3	5	0	3	0	7	5	2	0	3	0	5

The following code implements the sieve of Eratosthenes. The code assumes that each element of `sieve` is initially zero.

```
for (int x = 2; x <= n; x++) {
    if (sieve[x]) continue;
    for (int u = 2*x; u <= n; u += x) {
        sieve[u] = x;
    }
}
```

The inner loop of the algorithm is executed n/x times for each value of x . Thus, an upper bound for the running time of the algorithm is the harmonic sum

$$\sum_{x=2}^n n/x = n/2 + n/3 + n/4 + \dots + n/n = O(n \log n).$$

In fact, the algorithm is more efficient, because the inner loop will be executed only if the number x is prime. It can be shown that the running time of the algorithm is only $O(n \log \log n)$, a complexity very near to $O(n)$.

Euclid's algorithm

The **greatest common divisor** of numbers a and b , $\gcd(a, b)$, is the greatest number that divides both a and b , and the **least common multiple** of a and b , $\text{lcm}(a, b)$, is the smallest number that is divisible by both a and b . For example, $\gcd(24, 36) = 12$ and $\text{lcm}(24, 36) = 72$.

The greatest common divisor and the least common multiple are connected as follows:

$$\text{lcm}(a, b) = \frac{ab}{\gcd(a, b)}$$

Euclid's algorithm¹ provides an efficient way to find the greatest common divisor of two numbers. The algorithm is based on the following formula:

$$\gcd(a, b) = \begin{cases} a & b = 0 \\ \gcd(b, a \bmod b) & b \neq 0 \end{cases}$$

For example,

$$\gcd(24, 36) = \gcd(36, 24) = \gcd(24, 12) = \gcd(12, 0) = 12.$$

The algorithm can be implemented as follows:

```
int gcd(int a, int b) {
    if (b == 0) return a;
    return gcd(b, a%b);
}
```

It can be shown that Euclid's algorithm works in $O(\log n)$ time, where $n = \min(a, b)$. The worst case for the algorithm is the case when a and b are consecutive Fibonacci numbers. For example,

$$\gcd(13, 8) = \gcd(8, 5) = \gcd(5, 3) = \gcd(3, 2) = \gcd(2, 1) = \gcd(1, 0) = 1.$$

Euler's totient function

Numbers a and b are **coprime** if $\gcd(a, b) = 1$. **Euler's totient function** $\varphi(n)$ gives the number of coprime numbers to n between 1 and n . For example, $\varphi(12) = 4$, because 1, 5, 7 and 11 are coprime to 12.

The value of $\varphi(n)$ can be calculated from the prime factorization of n using the formula

$$\varphi(n) = \prod_{i=1}^k p_i^{\alpha_i - 1} (p_i - 1).$$

For example, $\varphi(12) = 2^1 \cdot (2 - 1) \cdot 3^0 \cdot (3 - 1) = 4$. Note that $\varphi(n) = n - 1$ if n is prime.

Modular arithmetic

In **modular arithmetic**, the set of numbers is limited so that only numbers $0, 1, 2, \dots, m - 1$ are used, where m is a constant. Each number x is represented by the number $x \bmod m$: the remainder after dividing x by m . For example, if $m = 17$, then 75 is represented by $75 \bmod 17 = 7$.

Often we can take remainders before doing calculations. In particular, the following formulas hold:

$$\begin{aligned} (x + y) \bmod m &= (x \bmod m + y \bmod m) \bmod m \\ (x - y) \bmod m &= (x \bmod m - y \bmod m) \bmod m \\ (x \cdot y) \bmod m &= (x \bmod m \cdot y \bmod m) \bmod m \\ x^n \bmod m &= (x \bmod m)^n \bmod m \end{aligned}$$

¹Euclid was a Greek mathematician who lived in about 300 BC. This is perhaps the first known algorithm in history.

Modular exponentiation

There is often need to efficiently calculate the value of $x^n \bmod m$. This can be done in $O(\log n)$ time using the following recursion:

$$x^n = \begin{cases} 1 & n = 0 \\ x^{n/2} \cdot x^{n/2} & n \text{ is even} \\ x^{n-1} \cdot x & n \text{ is odd} \end{cases}$$

It is important that in the case of an even n , the value of $x^{n/2}$ is calculated only once. This guarantees that the time complexity of the algorithm is $O(\log n)$, because n is always halved when it is even.

The following function calculates the value of $x^n \bmod m$:

```
int modpow(int x, int n, int m) {
    if (n == 0) return 1%m;
    long long u = modpow(x, n/2, m);
    u = (u*u)%m;
    if (n%2 == 1) u = (u*x)%m;
    return u;
}
```

Fermat's theorem and Euler's theorem

Fermat's theorem states that

$$x^{m-1} \bmod m = 1$$

when m is prime and x and m are coprime. This also yields

$$x^k \bmod m = x^{k \bmod (m-1)} \bmod m.$$

More generally, **Euler's theorem** states that

$$x^{\varphi(m)} \bmod m = 1$$

when x and m are coprime. Fermat's theorem follows from Euler's theorem, because if m is a prime, then $\varphi(m) = m - 1$.

Modular inverse

The inverse of x modulo m is a number x^{-1} such that

$$xx^{-1} \bmod m = 1.$$

For example, if $x = 6$ and $m = 17$, then $x^{-1} = 3$, because $6 \cdot 3 \bmod 17 = 1$.

Using modular inverses, we can divide numbers modulo m , because division by x corresponds to multiplication by x^{-1} . For example, to evaluate the value

of $36/6 \bmod 17$, we can use the formula $2 \cdot 3 \bmod 17$, because $36 \bmod 17 = 2$ and $6^{-1} \bmod 17 = 3$.

However, a modular inverse does not always exist. For example, if $x = 2$ and $m = 4$, the equation

$$xx^{-1} \bmod m = 1$$

cannot be solved, because all multiples of 2 are even and the remainder can never be 1 when $m = 4$. It turns out that the value of $x^{-1} \bmod m$ can be calculated exactly when x and m are coprime.

If a modular inverse exists, it can be calculated using the formula

$$x^{-1} = x^{\varphi(m)-1}.$$

If m is prime, the formula becomes

$$x^{-1} = x^{m-2}.$$

For example,

$$6^{-1} \bmod 17 = 6^{17-2} \bmod 17 = 3.$$

This formula allows us to efficiently calculate modular inverses using the modular exponentiation algorithm. The formula can be derived using Euler's theorem. First, the modular inverse should satisfy the following equation:

$$xx^{-1} \bmod m = 1.$$

On the other hand, according to Euler's theorem,

$$x^{\varphi(m)} \bmod m = xx^{\varphi(m)-1} \bmod m = 1,$$

so the numbers x^{-1} and $x^{\varphi(m)-1}$ are equal.

Computer arithmetic

In programming, unsigned integers are represented modulo 2^k , where k is the number of bits of the data type. A usual consequence of this is that a number wraps around if it becomes too large.

For example, in C++, numbers of type `unsigned int` are represented modulo 2^{32} . The following code declares an `unsigned int` variable whose value is 123456789. After this, the value will be multiplied by itself, and the result is $123456789^2 \bmod 2^{32} = 2537071545$.

```
unsigned int x = 123456789;
cout << x*x << "\n"; // 2537071545
```

Solving equations

Diophantine equations

A **Diophantine equation** is an equation of the form

$$ax + by = c,$$

where a , b and c are constants and the values of x and y should be found. Each number in the equation has to be an integer. For example, one solution for the equation $5x + 2y = 11$ is $x = 3$ and $y = -2$.

We can efficiently solve a Diophantine equation by using Euclid's algorithm. It turns out that we can extend Euclid's algorithm so that it will find numbers x and y that satisfy the following equation:

$$ax + by = \gcd(a, b)$$

A Diophantine equation can be solved if c is divisible by $\gcd(a, b)$, and otherwise it cannot be solved.

As an example, let us find numbers x and y that satisfy the following equation:

$$39x + 15y = 12$$

The equation can be solved, because $\gcd(39, 15) = 3$ and $3 \mid 12$. When Euclid's algorithm calculates the greatest common divisor of 39 and 15, it produces the following sequence of function calls:

$$\gcd(39, 15) = \gcd(15, 9) = \gcd(9, 6) = \gcd(6, 3) = \gcd(3, 0) = 3$$

This corresponds to the following equations:

$$\begin{aligned} 39 - 2 \cdot 15 &= 9 \\ 15 - 1 \cdot 9 &= 6 \\ 9 - 1 \cdot 6 &= 3 \end{aligned}$$

Using these equations, we can derive

$$39 \cdot 2 + 15 \cdot (-5) = 3$$

and by multiplying this by 4, the result is

$$39 \cdot 8 + 15 \cdot (-20) = 12,$$

so a solution to the equation is $x = 8$ and $y = -20$.

A solution to a Diophantine equation is not unique, because we can form an infinite number of solutions if we know one solution. If a pair (x, y) is a solution, then also all pairs

$$\left(x + \frac{kb}{\gcd(a, b)}, y - \frac{ka}{\gcd(a, b)}\right)$$

are solutions, where k is any integer.

Chinese remainder theorem

The **Chinese remainder theorem** solves a group of equations of the form

$$\begin{aligned}x &= a_1 \bmod m_1 \\x &= a_2 \bmod m_2 \\&\dots \\x &= a_n \bmod m_n\end{aligned}$$

where all pairs of m_1, m_2, \dots, m_n are coprime.

Let x_m^{-1} be the inverse of x modulo m , and

$$X_k = \frac{m_1 m_2 \cdots m_n}{m_k}.$$

Using this notation, a solution to the equations is

$$x = a_1 X_1 X_{1m_1}^{-1} + a_2 X_2 X_{2m_2}^{-1} + \cdots + a_n X_n X_{nm_n}^{-1}.$$

In this solution, for each $k = 1, 2, \dots, n$,

$$a_k X_k X_{km_k}^{-1} \bmod m_k = a_k,$$

because

$$X_k X_{km_k}^{-1} \bmod m_k = 1.$$

Since all other terms in the sum are divisible by m_k , they have no effect on the remainder, and $x \bmod m_k = a_k$.

For example, a solution for

$$\begin{aligned}x &= 3 \bmod 5 \\x &= 4 \bmod 7 \\x &= 2 \bmod 3\end{aligned}$$

is

$$3 \cdot 21 \cdot 1 + 4 \cdot 15 \cdot 1 + 2 \cdot 35 \cdot 2 = 263.$$

Once we have found a solution x , we can create an infinite number of other solutions, because all numbers of the form

$$x + m_1 m_2 \cdots m_n$$

are solutions.

Other results

Lagrange's theorem

Lagrange's theorem states that every positive integer can be represented as a sum of four squares, i.e., $a^2 + b^2 + c^2 + d^2$. For example, the number 123 can be represented as the sum $8^2 + 5^2 + 5^2 + 3^2$.

Zeckendorf's theorem

Zeckendorf's theorem states that every positive integer has a unique representation as a sum of Fibonacci numbers such that no two numbers are equal or consecutive Fibonacci numbers. For example, the number 74 can be represented as the sum $55 + 13 + 5 + 1$.

Pythagorean triples

A **Pythagorean triple** is a triple (a, b, c) that satisfies the Pythagorean theorem $a^2 + b^2 = c^2$, which means that there is a right triangle with side lengths a , b and c . For example, $(3, 4, 5)$ is a Pythagorean triple.

If (a, b, c) is a Pythagorean triple, all triples of the form (ka, kb, kc) are also Pythagorean triples where $k > 1$. A Pythagorean triple is *primitive* if a , b and c are coprime, and all Pythagorean triples can be constructed from primitive triples using a multiplier k .

Euclid's formula can be used to produce all primitive Pythagorean triples. Each such triple is of the form

$$(n^2 - m^2, 2nm, n^2 + m^2),$$

where $0 < m < n$, n and m are coprime and at least one of n and m is even. For example, when $m = 1$ and $n = 2$, the formula produces the smallest Pythagorean triple

$$(2^2 - 1^2, 2 \cdot 2 \cdot 1, 2^2 + 1^2) = (3, 4, 5).$$

Wilson's theorem

Wilson's theorem states that a number n is prime exactly when

$$(n - 1)! \bmod n = n - 1.$$

For example, the number 11 is prime, because

$$10! \bmod 11 = 10,$$

and the number 12 is not prime, because

$$11! \bmod 12 = 0 \neq 11.$$

Hence, Wilson's theorem can be used to find out whether a number is prime. However, in practice, the theorem cannot be applied to large values of n , because it is difficult to calculate values of $(n - 1)!$ when n is large.