Help You Participate

The International College Programming Contest

Lgarithm

November 11, 2013

Contents

Chapter 1

Preliminary

1.1 Introduction to ICPC

1.1.1 The ICPC: Problem, Thought and Balloon

The ACM/ICPC(International Collegiate Programming Contest, auspices by Association for Computing Machinery), is an annual contest which is popular among college students of computer science and some related majors.

Every year in October, November and December, regional contests will be held all over the world to select teams to advanced to the world final contest next year. Before the regional contests, many universities will hold their school contests, or subregional contests, also inviting teams from other universities to compete together.

As you can see in the ICPC logo, the contest is just like a game. Every three students form a team to solve problems using only one computer, once a problem is solved, they will be given a balloon with corresponding color.

Every problem has a simple, fast and wrong solution.

1.1.2 Traing Strategies: Make Use of Online Judges

There are many online judges(OJ) on the Internet, and They are all open and free. Anyone who wants to do practise can register for an account and submit solutions to judge. Here is a list of famous online judges:

- POJ, `http://poj.org/`
- ZOJ, `http://acm.zju.edu.cn/onlinejudge/`
- TJU, `http://acm.tju.edu.cn/`
- HDU, `http://acm.hdu.edu.cn/`

- SOJ, `http://cs.scu.edu.cn/soj/`
- SGU, `http://acm.sgu.ru/`
- URAL, `http://acm.timus.ru/`
- USACO, `http://ace.delos.com/usacogate`

We will use the shortcuts in this book. E.g. SOJ1001 means the problem on SOJ with ID 1001.

1.1.3 Setup Your Work Environment

The IDEs are: DEV C++, C-Free, Eclipse.

Install the Compiler

For windows users, I highly recomend you use the cygwin. You can download a strawberry perl, and use the gcc/g++ within it. `http://strawberryperl.com/`

Choose a Text Editor

Notepad++, gvim `http://notepad-plus-plus.org/` `http://www.vim.org/`

Start to Use the Colsole

Use stdio redirect to run test data. Use the `fc` commmand to check the answer.

Chapter 2

Number Theoty

Number theory is the most charming subject in mathematics. It deals problems about integers, which computer is good at. There are many problems on elementary number theory in the ICPC contest, each can be solved by beautiful solution with elegant coding skill.

2.1 Prime and Divisibility

Let m and n be integers, if exist $k \in \mathbb{Z}$ such that $m = nk$, we say m is a multiple of n and n is a divisor, or factor of m, denoted $n \mid m$.

For integers $m, n(n \neq 0)$, there is an unique pair of $r, q \in \mathbb{Z}$, such that $m = nq + r, 0 \leq r < |n|$, in the case $r \neq 0$, it is said to be the remindar of m divided by n. In C/C++ language, we use `m % n` to calculate the remindar of m divided by n, but when m is negative, the result r will be negative if it's now zero, and $-|n| < r < 0$. To make it positive, we should code as `(m % n + abs(n)) % n`, or usually `(m % n + n) % n` if provided n is positive.

A prime is a natural number that has no positive divisors other than 1 and itself. other numbers are composite, except 1, which is neither prime nor composite. The first primes are $2, 3, 5, 7, 11, \ldots$. There are 25 primes before 100, and 168 primes before 1000. There are infinite many primes and the number of primes no greater than n is approximately $n/\log n$.

2.1.1 Primality Test

For $n > 1$, if it can not be divided by any numbers between 1 and n, it must be a prime. But observe if n can be divide by $d(1 < d < n)$, it also can be dividey by n/d, either d or n/d must be smaller or equal to $\sqrt{n}$. So it's enough to check all numbers from 2 to $[\sqrt{n}]$. The worse case is when n is a prime or equals the square of a prime, but for random cases, this algoritm will soon end with a quite small divisor of n. In coding, we can use `i * i <= n` to avoid computing the square root of n.

```
int prime(int n)
{
    for (int i=2; i * i <= n; ++i)
        if (n % i == 0)
            return 0;
    return 1;
}
```

There are faster algorithms to do the primality test, such as Miller-Rabin test. Most of these algorithm are complicate, we don't disscuss them in this book. But we shall introduce the Eratosthenes's sieve, which generate a table of primes, and can be used when there is frequent primality tests of small numbers.

2.1.2 Eratosthenes's Sieve

For an integer $n > 1$, if we want to figure out all primes numbers from 1 to n, any algorithm that test each number will take a long time.

The Eratosthenes's sieve is a very old algorithm dates back to around 200 BC, it's idea is quite simple: for each numbers from 2 to n, if it is a prime, then mark all it's multiples other than itself to be composite, then find the next number which has not been marked yet, it must be a prime. Repeat the operation till the end. In fact, everytime we find a prime p, we only have to mark $p^2, p(p+1), p(p+2), \ldots p\left[\frac{n}{p}\right]$.

```
void eratosthenes(int n)
{
    for (int i=0; i <= n; ++i) p[i] = 1;
    for (int i=2; i <= n; ++i)
        if (p[i])
            for (int j=i * i; j <= n; j += i)
                p[j] = 0;
}
```

2.1.3 Factoring Numbers

An important fact which will be frequently used is: given an integer greater than 1, it can be decomposite into product of primes, and the decomposition is unique if we don't distinguish the order. This is know as the foundament theorem of arithmetic. In concrete term, for integer $n > 1$, we have

$$n = p_1^{e_1} \ldots p_k^{e_k}$$

where p_i are disditct primes and $e_i > 0$, this is called the standard factorization of n.

The solutions of many problems depend on the standard factorization of n. If we already know p is a prime factor of n, we can repeat divide n by p many

times until the qoutient can not be dividey by p any more. Suppose we divide n by p for k times, finally the quotient m can not divided by p, then $n = p^k m$, now we only have to factor m, which is smaller than n. So the problem of factoring number reduce to find a prime factor. Obviously this problem can not be easier than judge a number is prime. If we want to factor all numbers from 1 to n, inspired by the Eratosthenes's sieve, we can store a prime factor for each number during the process.

2.2 Linear Equation

2.2.1 $ax + by = c$

The equation $ax + by = c$, where a, b, c are integers served as a bridge links several aspects of number theory.

Theorem 2.2.1. *The equation $ax + by = c$ has a solution if and only if $\gcd(a, b) \mid c$.*

To prove this theorem, we need this lemma

Lemma 2.2.2. *The set $\{ax + by \mid x, y \in \mathbb{Z}\}$ equals to $\{kd \mid k \in \mathbb{Z}\}$ for some d.*

PROOF. *Choose $d \in \{ax+by \mid x, y \in \mathbb{Z}\}\backslash\{0\}$ with smallest absolute value, we assert $\{ax+by \mid x, y \in \mathbb{Z}\} = \{kd \mid k \in \mathbb{Z}\}$. Let $d = ax_1 + by_1$, for $c = ax_2 + by_2$, there is q, r such that $c = dq+r, (0 \le r < |d|)$, then $r = a(x_2 - x_1 q) + b(y_2 - y_1 q)$.*

Since $a, b \in \{ax + by \mid x, y \in \mathbb{Z}\}$, according to the lemma, $d \mid a$ and $d \mid b$, therefore $d \mid \gcd(a, b)$. On the other hand, for every common divisor e of a and b, $e \mid d$ because $d = ax_1 + by_1$. So $d = \gcd(a, b)$. □ Now we come to the conclusion that $\{ax + by \mid x, y \in \mathbb{Z}\}$ is the set of all multiples of $\gcd(a, b)$. This conclusion only guarantees the existence of solutions for $ax + by = d$, but we still don't know how to find one. So it's necessary to develop an algorithm to solve this problem.

Extended Euclid's Algorithm

Give a, b, the extended Euclid's algorithm finds a pair of (x, y) such that $ax + by = \gcd(a, b)$. Assume $a = bq + r$, if (x_1, y_1) is a solution of $ax + by = d$, then $(bq + r)x_1 + by_1 = d$, that is $b(y_1 + qx_1) + rx_1 = d$, therefore $(y_1 + qx_1, x_1)$ is a solution of $bx + ry = d$. If we already got a solution of $bx + ry = d$, which is (x_2, y_2), let $(y_1 + qx_1, x_1) = (x_2, y_2)$, we can figure out (x_1, y_1).

```
int euclid(int a, int b, int &x, int &y)
{
    if (b == 0)
    {
        x = 1;
        y = 0;
```

```
        return a;
    }
    else
    {
        int d = euclid(b, a % b, y, x);
        y -= a / b * x;
        return d;
    }
}
```

Euclid's algorithm

If we only wish to calculate the greatest common divisor of a and b, the above algorithm reduce to the classic Euclid's algorithm.

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

2.2.2 The Chinese Remainder Theorem

Theorem 2.2.3. *If $(m_i, m_j) = 1$ for all $i \neq j$, the linear congruence equations*

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \quad \vdots \\ x \equiv a_n \pmod{m_n} \end{cases}$$

has unique solution modulo $m_1 m_2 \dots m_n$.

The solution can be constructive. In fact $x = \sum a_i M_i M_i'$ is the solution, where $M_i = \frac{m_1 m_2 \cdots m_n}{m_i}$, M_i' is the inverse of M_i modulo $m_1 m_2 \dots m_n$.

If $O(m_1 + m_2 + \ldots + m_n)$ is bearable (in fact this is the most common case because $m_1 m_2 \dots m_n$ can't be too large), an enumerative algorithm can be carry out:

Let $x = 0$ initially, increase x by 1 each time until we got $x \equiv a_1 \pmod{m_1}$; then increase x by m_1 each time until we got $x \equiv a_2 \pmod{m_2}$; generally, in kth step, increase x by $m_1 \cdots m_{k-1}$ each time until we got $x \equiv a_k \pmod{m_k}$. Then we will got the solution after n steps.

2.3 Modular Arithmetic

Given an integer n, we say a congruence to b module n if n divides $a-b$ exactly, or $n \mid (a-b)$, and write $a \equiv b \pmod{n}$. The goodness of modular arithmetic is the number won't get very large.

2.3.1 The Repeated Squaring Method

To calculate $a^n \mod m$.

```
int mod_exp(int a, int n, int m)
{
    if (n == 1) return a;
    int tmp = mod_exp(a, n >> 1, m);
    tmp = (long long) tmp * tmp % m;
    if (n & 1) tmp = (long long) tmp * a % m;
    return tmp;
}
```

2.3.2 $ax = b \pmod{n}$

This equation can be transformed to $n \mid (ax - b)$, so there is an integer y, for which $ax - b = ny$, or equally $ax - ny = b$. So it has a solution if and only if $\gcd(a, n) \mid b$.

In particular, when $\gcd(a, n) = 1$ and $b = 1$, there is a number a' for which $aa' \equiv 1 \pmod{n}$, we call it the inverse of a module n.

Theorem 2.3.1 (Fermat-Euler)**.** *Let* $\gcd(a, n) = 1$*, then* $a^{\varphi(n)} \equiv 1 \pmod{n}$.

Fermat's little theorem is the special case when $n = p$, which is a prime.

2.3.3 Index and Primitive root

$\gcd(a, n) = 1$, the smallest d such that $a^d \equiv 1 \pmod{n}$ is called the index of a module n.

Theorem 2.3.2. *The index of a module n is a divisor of* $\varphi(n)$.

According to this theorem, if we want to calculate the index of a module n, we can enumerate all divisors of $\varphi(n)$, but we have more efficient ways.

First decomposite $\varphi(n)$ into $p_1^{e_1} \dots p_k^{e_k}$.

2.4 Carry System

2.5 Arithmetic Functions

Generally speacking, an arithmetic function is a function definded on integers only. And in ICPC, we only intrested in the arithmetic functions which take

values as integers too.

The common ones are $\phi(n), d(n), \sigma(n)$.

2.5.1 Möbius Transform

2.5.2 Dirichlet Convolution

2.6 Algebraic Methods

We introduce some algebraic approaches to make elementary number theory simpler. Beginners may skip this section.

2.6.1 Finite Abelian Group

Let $n > 1$ be an integer, the complete residue classes module n form a cyclic group $\mathbb{Z}/n\mathbb{Z}$, which is isomorphic to C_n, and the reduced residue classes module n form a finite Abelian group $(\mathbb{Z}/n\mathbb{Z})^*$.

When $n = 2^k$, we have

$$\begin{aligned}
(\mathbb{Z}/2\mathbb{Z})^* &\cong \{e\} \\
(\mathbb{Z}/4\mathbb{Z})^* &\cong C_2 \\
(\mathbb{Z}/8\mathbb{Z})^* &\cong C_2 \oplus C_2 \\
(\mathbb{Z}/16\mathbb{Z})^* &\cong C_2 \oplus C_4 \\
(\mathbb{Z}/32\mathbb{Z})^* &\cong C_2 \oplus C_8 \\
&\vdots \\
(\mathbb{Z}/2^k\mathbb{Z})^* &\cong C_2 \oplus C_{k-2}
\end{aligned}$$

When $n = p^k$, where p is an odd prime, we know n do have primitive roots according to elementary number theory, thus $\mathbb{Z}/n\mathbb{Z}$ is a cyclic group of order $\varphi(p^k) = p^k - p^{k-1}$.

Chapter 3

Combinatorics

3.1 Permutation

3.1.1 Group of Permutations

3.1.2 number of permutations and set partitions of a given shape

We say a permutation is of shape

$$1^{\lambda 1}2^{\lambda 2}\dots n^{\lambda n}$$

if it has exactly λ_i cycles of length i. And we also say a partition of a set is of shape 3.1.2 if it has exactly λ_i subsets of cardinate i. The number of permutations and set partitions of shape (1) are well know as

$$\frac{n!}{\lambda_1!\lambda_2!\dots\lambda_n!1^{\lambda_1}2^{\lambda_2}\dots n^{\lambda_n}}$$

and

$$\frac{n!}{\lambda_1!\lambda_2!\dots\lambda_n!(1!)^{\lambda_1}(2!)^{\lambda_2}\dots (n!)^{\lambda_n}}$$

respectively.

3.2 Counting

Enumerative combinatorics counts the cardinate of a finite set. Let X, A and B be finite sets, then $|A \times B| = |A| \times |B|$. And $|B^A|$, the number of functions $A \to B$ is $|B|^{|A|}$, because $\forall a \in A$, $f(a)$ has $|B|$ choices, and the choices are independent. The special case is $\{0,1\}^X = 2^{|X|}$. Recall $|\mathcal{P}(X)| = 2^{|X|}$ because every element must be in or not in a specific subset. In fact, there is a one-to-one

corresponding between $\mathcal{P}(X)$ and $\{0,1\}^X$ [1], $A \to \chi_A$, $\chi_A(x) = \begin{cases} 1, x \in A \\ 0, x \notin A \end{cases}$. Since χ_A determines A uniquely, we call it the character function of A.

3.2.1 Elementary Counting

3.2.2 Inclusion-Exclusion Principle

Let A_n be a finite family of subsets of A, define $C(x) = \sum_{i=1}^{n} \chi_{A_n}(x)$, we have sieve formula: $\#\{x \mid C(x) = 0\} = \#A - \sum \#A_i + \sum \#(A_i \cap A_j) + \cdots + (-1)^r \sum \#(A_{i_1} \cap \cdots \cap A_{i_r}) + \cdots + (-1)^n \sum \#(A_{i_1} \cap \cdots \cap A_{i_n})$. More generally, we have: $\#\{x \mid C(x) = k\} = \sum_{r=k}^{n} \binom{r}{k} (-1)^{r+k} \sum \#(A_{i_1} \cap \cdots \cap A_{i_r})$. The proof is based on the equation $\sum_{r=k}^{m} \binom{r}{k}\binom{m}{r}(-1)^r = \delta(m,k)$.

3.2.3 Pólya Enumeration Theorem

Let group G acts on set X, then

$$\#O(x) = [G : G_x].$$

Let N be the number of oribits, we have

$$N = \sum \frac{1}{|O(x)|} = \frac{1}{|G|} \sum |G_x| = \frac{1}{|G|} \sum fix(g).$$

That is, the number of orbits equals the average number of fixed points.

3.2.4 Counting Geometric Shapes

Euler's formula $V - E + F = 2$

Euler's formula for planar graph is a very beautiful formula, we shall use it to a class of counting problems. There are three variables in the equation, only we know two of them, the third can be calculated. [2]

A basic example is to calculate how many parts can n lines cut the plane if no two lines parallel to eath other and no three lines intersect at a point. To calculate it, we must translate it into a finite planar graph by adding a very large circle containing all intersection points first. Since there are $V = \binom{n}{2} + n = \frac{n(n-1)}{2} + n$ vertices and $E = n^2 + n$ edges, $F = E - V + 2 = \frac{n(n+1)}{2} + 2$, but we only count the number of faces inside the circle, so the answer is $\frac{n(n+1)}{2} + 1$.

Pick's theorem

Another class of counting problems is related to two dimensional lattice. Say counting the number of points whose coordinates are integers inside a shape.

[1] that's why we also write 2^X for $\mathcal{P}(X)$.

[2] sample problem: Beijing 2004 Preliminary, Fourier's Lines

To do this, we need a lemma. If the coordinates of a triangle's three vertices are integers, and there is no other point inside or on the edges of that triangle whose coordinates are integers, then the area of the triangle is $\frac{1}{2}$. We can create a planar graph by connect as many as possible points whose coordinateds are integers inside or en edges of that shage, and no two segments intersec properly, then the number of small triangles is equal to 2 times the area of that shape. Then we calculate the total degree of all small triangles in two ways. Every inside point contributes 2π degree and every point on edges except vertices contributes π, the total degree contribute by vertices is $(n-2)\pi$. let i be the number of inside points, j be the number of points on boundary except vertices, n be the number of vertices, A be the area, we have $2A\pi = 2i\pi + j\pi + (n-2)\pi$, that is $A = i + \frac{j+n}{2} - 1$, or $A = i + \frac{b}{2} - 1$, b is the number of points on doundary.

3.3 Generating Function

3.3.1 Formal Power Series

Generating function is a power tool to describe sequence. If $\{a_n\}$ is a sequence, $\sum a_n x^n$ is its generating function. Since a sequence can be infinite, the generating function is not a polynomial, but a formal series(by formal means we don't consider the convergence). The sum of two formal series is defined by pointwise sum of their coefficients, i.e. $\sum a_n + \sum b_n = \sum(a_n + b_n)$. The product of two formal series is defined by the convolution of their coefficients, i.e. $(\sum a_n)(\sum b_n) = \sum(\sum_{k=0}^{n} a_k b_{n-k})$.

Some commom generating functions are:

$$\frac{1}{1-x} = 1 + x + x^2 + \ldots + x^n + \ldots$$

$$\frac{1}{(1-x)^{1+n}} = \binom{n}{n} + \binom{n+1}{n}x + \binom{n+2}{n}x^2 + \ldots + \binom{n+k}{n}x^k + \ldots$$

Another kind of generating function of $\{a_n\}$ is $\sum \frac{a_n}{n!}x^n$.

3.3.2 Close Form

3.4 Combinatorics Numbers

binomial coefficient

The binomial coefficients, $\binom{n}{k}$ are coefficients of x in the expansion $(1+x)^n$. More precisely, $(1+n)^n = \sum_{k=0}^{n} \binom{n}{k} x^k$.

Catlan number

Many enumeration problem has answer C_n, where C_n has the induction formula

$$C_{n+1} = \sum_{k=0}^{n} C_k C_{n-k}, C_0 = 1.$$

The first several Catalan numbers are: $C_0 = 1$, $C_1 = 1$, $C_2 = 2$, $C_3 = 5$, $C_4 = 14$, $C_5 = 42$.

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

partition number

Let $p(n)$ denote the number of ways to write n into sums of positive numbers, regardless of order, and $p(n,k)$ denote the number of ways when there are exactly k positive numbers, we make the convention $p(0,0) = 1$ and $p(n,k) = 0$ when $n < k$ for convenience. The partition of the first five positive numbers are listed below:
1
$2 = 1 + 1$
$3 = 2 + 1 = 1 + 1 + 1$
$4 = 3 + 1 = 2 + 2 = 2 + 1 + 1 = 1 + 1 + 1 + 1$
$5 = 4 + 1 = 3 + 2 = 3 + 1 + 1 = 2 + 2 + 1 = 2 + 1 + 1 + 1 = 1 + 1 + 1 + 1 + 1$
we can see $p(1) = 1$, $p(2) = 2$, $p(3) = 3$, $p(4) = 5$, $p(5) = 7$ and
$p(1,1) = 1$,
$p(2,1) = 1$, $p(2,2) = 1$,
$p(3,1) = 1$, $p(3,2) = 1$, $p(3,3) = 1$,
$p(4,1) = 1$, $p(4,2) = 2$, $p(4,3) = 1$, $p(4,4) = 1$,
$p(5,1) = 1$, $p(5,2) = 2$, $p(5,3) = 2$, $p(5,4) = 1$, $p(5,5) = 1$,
from above. A partition of n into k parts has two situations, one is each summand is greater than one, in this situation, we subtract 1 from each summand and obtain a partion of $n-k$ into k parts, another situation is there is at least one 1 in the summand, in this situation, we eliminate a 1 and obtain a partition of $(n-1)$ into $(k-1)$ parts. Hence we got the induction formula

$$p(n,k) = p(n-1,k-1) + p(n-k,k)$$

therefore $p(n) = \sum_{k=0}^{n} p(n,k)$.

Observe the value of $p(n,k)$ for small n and k futher more, we found $p(n,1) = p(n,n) = 1(n > 0)$, $p(n,2) = [\frac{n}{2}]$, $p(n,n-1) = 1(n > 1)$, $p(n,n-2) = 2(n > 3)$, and $p(n,k) = P(n-1,k-1)$ when $2k > n$, becasue $p(n-k,k) = 0$ when $n - k < k$. Let $n = r + k + 1$

Stirling number of the second kind and Bell number

Let $S(n,k)$ denote the number of ways to divide a set of n elements into k nonempty disjoint parts. Define $S(n,k) = 0$ when $n < k$, $S(n,0) = 0$ when $n > 0$

and $S(0,0) = 1$. Obviously $S(n,1) = S(n,n) = 1$.
Example:
all partitions of $\{1,2,3\}$ are:
$\{1,2,3\}$,
$\{1\} \cup \{2,3\}$, $\{2\} \cup \{1,3\}$, $\{3\} \cup \{1,2\}$,
$\{1\} \cup \{2\} \cup \{3\}$,
therefore $S(3,2) = 3$;
all partitions of of $\{1,2,3,4\}$ are:
$\{1,2,3,4\}$,
$\{1\} \cup \{2,3,4\}$, $\{2\} \cup \{1,3,4\}$, $\{3\} \cup \{1,2,4\}$, $\{4\} \cup \{1,2,3\}$,
$\{1,2\} \cup \{3,4\}$, $\{1,3\} \cup \{2,4\}$, $\{1,4\} \cup \{2,3\}$,
$\{1\} \cup \{2\} \cup \{3,4\}$, $\{1\} \cup \{3\} \cup \{2,4\}$,
$\{1\} \cup \{4\} \cup \{2,3\}$, $\{2\} \cup \{3\} \cup \{1,4\}$,
$\{2\} \cup \{4\} \cup \{1,3\}$, $\{3\} \cup \{4\} \cup \{1,2\}$,
$\{1\} \cup \{2\} \cup \{3\} \cup \{4\}$,
therefore $S(4,2) = 7$, $S(4,3) = 6$. We list the value of $S(n,k)$ for $1 \leq k \leq n \leq 4$ below:
$S(1,1) = 1$,
$S(2,1) = 1$, $S(2,2) = 1$,
$S(3,1) = 1$, $S(3,2) = 3$, $S(3,3) = 1$,
$S(4,1) = 1$, $S(4,2) = 7$, $S(4,3) = 6$, $S(4,4) = 1$.
Consider a partion of $\{1,2,\ldots,n\}$ into k parts, $\{1,2,\ldots,n\} = U_1 \cup U_2 \cdots \cup U_k$. If $\{n\} = U_i$ for some i, then the reset subsets form a partition of $\{1,2,\ldots,n-1\}$ into $k-1$ parts, otherwise, we eliminate n form the subset it belongs, that subset still remains non-empty, thus we got a partition of $\{1,2,\ldots,n-1\}$ into k parts. But given a partition of $\{1,2,\ldots,n-1\}$ into k parts, we have k ways to add n into one of the k subsets, therefore we got the induction formula for $S(n,k)$, which is

$$S(n,k) = S(n-1,k-1) + kS(n-1,k).$$

The sum $B(n) = \sum_{k=0}^{n} S(n,k)$ is know as Bell number, which denotes the number of all partitions of a set with n elements. $B(n)$ has a recursive formula

$$B(n+1) = \sum_{k=0}^{n} \binom{n}{k} B(k).$$

3.5 Advanced Topics

3.5.1 Poset

If there is a partial order $\leq$ on a set P, we say $(P, \leq)$, or simply P is a poset. A chain on a poset is a sequence $\{a_n\}$ such that $a_1 \leq a_2 \leq ... \leq a_n$ and no two elements are equal, n is called the length of the chain. An anti-chain is a subset A of P such that no two elements in that subset are compareble, $\#A$ is called the length of the anti-chain.

The Dilworths' theorem and its dual theorem say: the maximum length of a chain is equal to the minimum number of anti-chain cover; the maximum length of an anti-chain is equal the mimimum number of chain cover.

3.5.2 Incidente Algebra

matrix and vector on poset

A n dimensional vector $v(v_1, v_2, \ldots, v_n)$ on K can be viewed as a function

$$v : [n] \to K,$$

while a n by n matrix $[M_{ij}]$ on K can be viewed as a function

$$M : [n] \times [n] \to K.$$

Now we consider $[n]$ to be any another poset.

Let P be a local bound poset, and α, β be matrix, i.e.

$$\alpha, \beta : P \times P \to K,$$

define the product of α and β by

$$(\alpha\beta)(x, y) = \sum_{x \le z \le y} \alpha(x, z)\beta(z, y).$$

If we restrict the entry $\alpha(x, y)$ in the matrix to be non-zero only when $x \le y$, the the product we've defined agrees the product of two matrix. Let f, g be vectors, i.e.

$$f, g : P \to K,$$

define the product of vector and matrix by

$$(\alpha f)(x) = \sum_{x \le y} \alpha(x, y) f(y)$$

and

$$(f\alpha)(x) = \sum_{y \le x} f(y)\alpha(y, x)$$

vector can be either row vector or column vecotr, depend on the matrix multiply it is on the left or right. Let

$$I : P \times P \to K, I(x, y) = \begin{cases} 1 & x \le y \\ 0 & x \not\le y \end{cases}$$

then

$$f(x) = \sum_{y \le x} g(y)$$

can be write in the form

$$f = gI,$$

while

$$f(x) = \sum_{x \leq y} g(y)$$

can be write in the form

$$f = Ig.$$

Möbius inversion formula

Let μ be the inverse of I such that

$$\mu I = I\mu = \delta,$$

where

$$\delta(x, y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}.$$

then

$$\delta(x, y) = \sum_{x \leq z \leq y} \mu(x, z) I(z, y) = \sum_{x \leq z \leq y} \mu(x, z)$$

therefore

$$\mu(x, x) = 1$$

and

$$\sum_{x \leq z \leq y} \mu(x, z) = 0, x \neq y$$

Also we have

$$\delta(x, y) = \sum_{x \leq z \leq y} I(x, z)\mu(z, y) = \sum_{x \leq z \leq y} \mu(z, y),$$

and

$$\sum_{x \leq z \leq y} \mu(z, y) = 0, x \neq y$$

Now when we know $f = gI$ and $f = Ih$, we can calculate g and h as

$$g(x) = (f\mu)(x) = \sum_{y \leq x} f(y)\mu(y, x)$$

and

$$h(x) = (\mu f)(x) = \sum_{x \leq y} \mu(x, y) f(y).$$

3.6 Constructive Combinatorics

3.7 String

3.7.1 The KMP Algorithm

3.7.2 Trie and Aho-Corasick Algorithm

3.7.3 Introdution to Automata and Language

Chapter 4

Geometry

4.1 Point and Line

We use coordinates to represent point and linear equations to represent lines. The equation of the line passes $P(x_1, y_1)$ and $Q(x_2, y_2)$ is

$$\begin{vmatrix} X & Y & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} = 0,$$

or equally

$$\begin{vmatrix} y_1 & 1 \\ y_2 & 1 \end{vmatrix} X - \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix} Y + \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} = 0,$$

and the intersection point of line $L(X, Y) = a_1X + b_1Y + c_1 = 0$ and $M(X, Y) = a_2X + b_2Y + c_2 = 0$ is

$$(\frac{\begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix}}{d}, -\frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{d}),$$

where $d = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}$. If we use homogeneous coordinates for points, the intersection point will be

$$(\begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix}, -\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}, \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}).$$

4.2 Convex Hull

Chapter 5

Graph

5.1 Basic Concepts

A graph is a set of vertices connected by edges. A graph G, denoted by $G(V, E)$, V is called the set of vertex and E is called the set of edges.

When all edges connect different vertices, and every two edges is connected by at most one edge, we call the graph a simple graph.

Usually, we assume the edge has no direction, in that case, E can be viewed as an irreflexive and symmetric binary realtion V.

If the edges has directed, that is, there may be a edge from u to v but no edge from v to u. The relation E on V is not symmetric any more. But we assume for every two edges u and v, there is at most one edge from u to v, and there is no edge from u to u. Then we can still use a irreflexive binary relation on V to model a graph.

But sometimes, there may be edges from a vertex to itself, and there may be more than one edges between two vertices, we call the graph a multiple graph. A multiple graph can not be modeled by binary relations. So we give the formal definitions:

An undirected multiple graph $G(V, E)$ is a set V of vertices and a set E of edges, and there exist a function associate each edge with two vertices.

A directed multiple grapg $G(V, E)$ is a set V of vertices and a set E of edges, and there exist two functions $s, e : E \to V$, for each edge $l \in E$, $s(l)$ is called the start of l, $e(l)$ is called the end of l.

5.2 Simple Graph

5.3 Shortest Path

There are two kinds of problems, SSSP and APSP.

5.3.1 Signal Source Shortest Path(SSSP)

5.3.2 All Pair Shortest Path(APSP)

5.4 Network Flow

By a network, we mean a positive weighted directed multiple graph $G(V, E)$, for each $e \in E$, the weight $c(e) > 0$ of e is called the capacity of e. A flow on G is a function $f : E \to \mathbb{R}^+ \cup \{0\}$. For a vertices $v \in V$, let $f^-(v) = \sum_{(u,v)\in E} f(u, v)$ and $f^+(v) = \sum_{(v,u)\in E} f(v, u)$.

If given two distinct vertices s and t in V, a flow f on G is a proper flow from s to t if $f(e) \leq c(e)$ for all $e \in E$ and $f^+(v) = f^-(v)$ for all $v \in V\backslash\{s, t\}$.

Proposition 5.4.1. *If f is a proper flow from s to t, then $f^+(s) = f^-(t)$.*

Let f, g be two proper flows on G from s to t, $s, t \in V, s \neq t$.

5.5 Bipart Graph

5.5.1 The Hungary Algorithm

5.5.2 The KM Algorithm

Chapter 6

Game Theory

6.1 Game of Nim

The game of nim is a game played by two players, who alternatively take turns to remove stones from a pile among several. The one who can not move any stones lose. If there are n piles of stones, a game of nim can be represent by n numbers $x_1, x_2, \ldots, x_n$.

Theorem 6.1.1. *The player move first has a winning strategy if and only if*

$$x_1 \oplus x_2 \oplus \cdots x_n > 0.^1$$

PROOF. *Notice the two facts:*

Suppose $x = x_1 \oplus \cdots \oplus x_n$, then one of the x_i changes its value will cause the change of x. Assume we x_1 change to $y_1 (y_1 \neq x_1)$, then $y_1 \oplus x_2 \oplus \cdots x_n = y_1 \oplus x_1 \oplus x_1 \oplus x_2 \oplus \cdots \oplus x_n = y_1 \oplus x_1 \oplus x$. hence $y_1 \neq x_1$, therefore $y_1 \oplus x_1 \neq 0$, $y_1 \oplus x_1 \oplus x \neq x$.

Another fact is, if $x = x_1 \oplus \cdots \oplus x_n > 0$, there exists a move which changes x_i to $y_i (y_i < x_i)$, will make x zero.

If such move exists, we have, assume the index i is 1 without lose of generality, then $0 = y_1 \oplus x_2 \oplus \cdots x_n = y_1 \oplus (x_1 \oplus x_1) \oplus \cdots x_n = (y_1 \oplus x_1) \oplus (x_1 \oplus \cdots x_n)$, therefore $x_1 \oplus y_1 \oplus x = 0$ and $y_1 = x \oplus x_1$. We must prove such x_i and $y_i < x_i$ exists.

In fact, there must be at least one digital 1 is the binary experssion of a non zero number. Suppose $x = 2^{d_1} + 2^{d_2} + \cdots + 2^{d_k}, d_1 > d_2 > \cdots > d_k$, so the ones are in the d_ith position of the binary expression of x, therefore at least one of x_i must have a one in d_1th position, turn the digitals in the d_1th, d_2th, $\ldots$, d_kth position of that number to opposite state then that number must becomes smaller, since the highest bit we change is from 1 to 0. Meanwhile x becomes zero.

[1] $\oplus$ means exclusive or.

6.1.1 Simplify Rules

Sometimes a game is hard to analysis directly, so we must transform it to another simpler case. There are two rules to do this:

If there are two equivalent state in a game, they can be eliminated simultaneously.

If there is a state which makes the first player lose, it can be eliminated from the game.

6.2 Combinatory Game

Chapter 7

Best Strategy

7.1 Dynamic Programming

7.1.1 The Longest Increasing Subsequence(LIS) Problem

7.1.2 Knapsack Problem

Chapter 8

Search

8.1 Floodfill

8.2 BFS

8.3 DFS

Chapter 9

Numeric

9.1 Linear Algebra

9.1.1 Determinant

Evulate a matrix.

9.1.2 Linear Equations

9.2 Discrete Fourier Transform

Let P and G be two polynomials of degree n, if we want to calculate their product, it will take $O(n^2)$.

Let $\{a_k\}_{k=0}^{n-1}$ be a sequence of n numbers, the discrete Fourier transform of it is another sequence $\{x_k\}_{k=0}^{n-1}$ defined by

$$x_k = \sum_{r=0}^{n-1} a_r \exp(\frac{2\pi ikr}{n})$$

and the inverse transform is given by

$$a_k = \frac{1}{n} \sum_{r=0}^{n-1} x_r \exp(-\frac{2\pi ikr}{n})$$

Let $\{a_k\}$ and $\{b_k\}$ be two sequences of length n, the circular colvolution of them is anther sequence $\{c_k\}$ defined by

$$c_k = \sum_{r+s \equiv k \pmod n} a_r b_s$$

9.2.1 Circular Convolution

The discrete Fourier transform has something to to with circular convolution, that is

$$a \circ b = \mathcal{F}^{-1}(\mathcal{F}(a) \cdot \mathcal{F}(b))$$

If we can calculate $\mathcal{F}(a)$ and $\mathcal{F}^{-1}(a)$ in $O(n \log n)$ time, then multiply polynomials will be also in $O(n^2)$ time.

9.2.2 Fast Fourier Transform

To do fast Fourier transform, n must be a power of 2, if n is not a power of 2, we can choose an integer wich is a power of 2 and greater than n, then add some padding zeros at the end of the sequence.

Let $w_N = e^{-\frac{2\pi i}{N}}$, a primitive N-th unit root. When N is even, $w_{N/2} = w_N^2$. Let $x = (x_n)$ be a sequence of length N, $p_0, p_1, \ldots, p_{N/2}$ be the DFT of $x_0, x_2, \ldots, x_{N-2}$, $q_0, q_1, \ldots, q_{N/2}$ be the DFT of $x_1, x_3, \ldots, x_{N-1}$, y be the DFT of x. Then we have

$$p_n = \sum_{k=0}^{N/2-1} x_{2k} w_{N/2}^{nk} \quad n = 0, 1, \ldots, N/2 - 1$$

$$q_n = \sum_{k=0}^{N/2-1} x_{2k+1} w_{N/2}^{nk} \quad n = 0, 1, \ldots, N/2 - 1$$

and

$$\begin{aligned} y_n &= \sum_{k=0}^{N-1} x_k w_N^{nk} \quad n = 0, 1, \ldots, N-1 \\ &= \sum_{k=0}^{N/2-1} x_{2k} w_N^{2nk} + \sum_{k=0}^{N/2-1} x_{2k+1} w_N^{n(2k+1)} \\ &= \sum_{k=0}^{N/2-1} x_{2k} w_{N/2}^{nk} + w_N^n \sum_{k=0}^{N/2-1} x_{2k+1} w_{N/2}^{nk} \end{aligned}$$

therefore, when $0 \le n < N/2$,

$$y_n = p_n + w_N^n q_n$$

and because of $w_{N/2}^{nk} = w_{N/2}^{(n+N/2)k}$, $w_N^{N/2} = -1$,

$$y_{n+N} = p_n + w_N^{n+N/2} q_n = p_n - w_N^n q_n.$$

According to this relation, if p and q are known, we can calculate y only in $O(N)$ time, but this requires N to be even. If $N = 2^l$, p and q also can be calculated in the same way. So we obtained a recursive algorithm to calculate DFT, using

only $O(N \log N)$ time, as long as $N = 2^l$. The time complexity can be proved by induction, when $N = 1$, we only let $y_0 = x_0$; when $N = 2^l$, we can calculate both p and q in

$$O(\frac{N}{2} \log \frac{N}{2})$$

time, so the total time is

$$O(\frac{N}{2} \log \frac{N}{2}) + O(\frac{N}{2} \log \frac{N}{2}) + O(N) = O(N \log N)$$

Here gives the main process of FFT.

9.2.3 Number Theoretic Transform

Instead of $\mathbb{C}$, we can perform DFT over a finite field, an N-th primitive root will be used in place of w_N. It's most convenient to take the finte field $\mathbb{Z}/p\mathbb{Z}$, where p is a prime. If $p = 2^l k + 1$, then $\mathbb{Z}/p\mathbb{Z}$ contains 2^l-th primitive roots. According to Dirichlet's theorem on arithmetic progress, given l fixed, there are infinite many primes such that $p - 1$ divides 2^l. In the range of 32-bit integers, $2013265921 = 2^{27} * 15 + 1$ is a good choice.

Appendix

A A list of problems about number theory on SOJ

1012 Fermat vs_ Pythagoras
1020 Triangle
1022 Uniform Generator
1023 Prime Cuts
1059 Pi
1088
1092
1115
1148 Secret Code
1159 Factorial
1630
1743 How Many Common Divisors?
1748 How Many Common Divisors?(II)
1763 Sum of Factorials
1764 Vivian's Problem
1802 Quadratic Residues
1933 Looooops
1974 Mod 9
2011 Ones
2087 Happy 2005
2115
2191 Farey Sequence
2193 Longge's problem
2326 Factorial Factors
2471 Reduced ID Numbers
2478 Simple Task
2479 Simple Task II
2498 Count prime
2501 Bedtime Reading, I
2502 Bedtime Reading, II
2577 Prime Path
2583 Sum of Different Primes
2661 Farey Sequence
2666 n!
2668 C(n,k)
2704 Combination
2763 Factorial
2833 Euler's totient function
2843 Self Numbers
2858 Happy 2006
2880 Semi-prime H-numbers
2911 Divisors
3006 Last digit
3019 Count Color
3020 X-factor Chains
3137 Simple Computing
3138 Simple Computing II
3165 Big^n
3173 Factor count
3249 Number Sets
3252 Choose
3273 Count the factors
3274 Goldbach's Conjecture
3290 Distribute The Apples
3291 Distribute The Apples II
3293 Rescue Me I
3294 Rescue Me II
3298 Divisibility
3315 Natasha's Problem
3416 YAPTCHA
3450 Divisors showdown
3451 Euler's totient function
3595 Product of Divisors

Bibliography

[1] An Introduction to the Theory of Numbers, G.H.Hardy, E.M.Wright

[2] Friendly Introduction to Number Theory, Joseph H. Silverman

[3] Elementary Number Theory and Its Applications, Kenneth H. Rosen

[4] Basic Notations of Algebra, Igor R. Shafarevich

[5] A First Course in Abstract Algebra with Application, Joseph J. Rotman

[6] Enumerative Combinatorics, Volume I, Richard P.Stanley

[7] A Course in Combinatorics, J.H.van Lint, R.M.Wilson