离散数学 for CS

spiritTrance

2023 年 4 月 4 日

前言

笔记基本信息

从2023年4月3日开始做此笔记。

作者的话

考虑到本校所教离散数学的内容稍有不全（数论及形式语言自动机），以及本人需要复习（抽象代数），所以有此篇笔记。考虑到离散数学对于计算机学科是相当重要的基础课程，因此本篇笔记会穿插一定的代码实现（**~~其实是临时学了数论想试一下~~**），同时会在合适处给出例题链接。

参考教材

大名鼎鼎的黑皮书以及潘承洞老师的初等数论。

笔记作者联系方式

略。

警告

本笔记主要还是供复习使用，如果需要更详细的解释，请查阅相关资料。

目录

1 初等数论初步

> “初等”的“初步”，看起来算简单了。——佚名

1.1 整除及同余理论

1.1.1 整除

对于整数$n > 0, q > 0$，当$q \neq 0$时总能找到整数p和$r(0 \leq r < q)$，满足式(1)：

$$n = pq + r \tag{1}$$

当r为0时，我们称q整除n，记为$q|n$；当$r \neq 0$时，我们记$r = n \mod q$.

1.1.2 最大公约数，最小公倍数

考虑整数a, b，其最大公约数记为(a, b)，最小公倍数记为$[a, b]$，则有式(2)成立。

$$(a, b)[a, b] = ab \tag{2}$$

1.1.3 算术基本定理

定理1.1.1 算术基本定理

对于$\forall n \geq 1, n \in \mathbb{Z}$，有式(3)成立：

$$n = p_1^{c_1} p_2^{c_2} p_3^{c_3} \dots p_s^{c_s} \tag{3}$$

其中$p_i, i \in [1, s]$为不大于n的所有质数，且$p_i < p_j, i < j$，c_i为非负整数。

这个定理好像显然。考虑$a = p_1^{m_1} p_2^{m_2} p_3^{m_3} \dots p_s^{m_s}$ 和$b = p_1^{n_1} p_2^{n_2} p_3^{n_3} \dots p_s^{n_s}$. 因此，我们有式(4)和(5)成立：

$$(a, b) = p_1^{\delta_1} p_2^{\delta_2} p_3^{\delta_3} \dots p_s^{\delta_s}, \delta_i = \min(m_i, n_i) \tag{4}$$

$$[a, b] = p_1^{\gamma_1} p_2^{\gamma_2} p_3^{\gamma_3} \dots p_s^{\gamma_s}, \gamma_i = \max(m_i, n_i) \tag{5}$$

有了这个定理后，我们有以下式子成立，有兴趣的可以自己证明。

$$(a,b) = (a+nb,b) = \frac{1}{m}(ma,mb)$$
$$[a,b] = \frac{1}{m}[ma,mb]$$
$$[a_1,a_2,\dots a_i,a_{i+1},\dots,a_n] = [[a_1,a_2,\dots a_i],[a_{i+1},\dots,a_n]]$$
$$(a_1,a_2,\dots a_i,a_{i+1},\dots,a_n) = ((a_1,a_2,\dots a_i),(a_{i+1},\dots,a_n))$$
$$n = pq + r \Leftrightarrow (n,q) = (r,q)$$
(Bezout定理) $$(a,b) = ax_0 + by_0 = \inf_{x_0,y_0}\{ax_0+by_0 > 0 | x_0,y_0 \in \mathbb{Z}, a,b\text{不全为}0\}$$
$$(a,c) = 1 \Rightarrow (ab,c) = (b,c)$$
$$(a,b) = 1 = (ab,d) = (a,d)(b,d)$$
$$(a,c) = 1, c|ab \Rightarrow c|b$$

1.1.4 辗转相除法与一次不定方程

```
int gcd(int a, int b){
    return a%b ? gcd(b, a%b): b;     //注意a与b的大小关系可以无关
}
```

```
int exgcd(int a, int b, int &x, int &y){
    if (b == 0){
        x = 1; y = 0; return a;
    }
    int d = exgcd(b, a%b, x, y);
    int z = x; x = y; y = z-y * (a / b);
    return d;
}
```

1.1.5 [x]与{x}函数

1.2 数论函数

1.2.1 常见数论函数

定义1.2.1 欧拉函数

欧拉函数$\varphi(x)$表示与x互素的不大于x的自然数个数，其计算方式如(6)所示（**如何证明？**[a]）：

$$\varphi(x) = x\prod_{i=1}^{s}(1-\frac{1}{p_i}) \tag{6}$$

其中$x = p_1^{c_1}p_2^{c_2}p_3^{c_3}\dots p_s^{c_s}(c_i \neq 0, p_i)$是不大于$x$的质数。

[a] tips:筛质数，用容斥原理

欧拉函数具有以下性质[1]：

性质1 $\varphi(p) = p-1$，其中p是质数

[1] 务必思考这些性质的证明。

性质2 $\sum_{(d,n)=1} d = \frac{n\varphi(n)}{2}$

性质3 $\sum_{d|n} \varphi(d) = n$

性质4 当p为质数且$p \mid n$时，若$p^2|n$，有$\varphi(n) = \varphi(n/p) * p$，否则有$\varphi(n) = \varphi(n/p) * (p-1)^2$

```
int phi(int n){
    int ans = n;
    for (int i = 2; i <= sqrt(n); i++){
        if (n % i == 0){
            ans = ans / i * (i - 1);
            while(n % i == 0) n /= i;
        }
    }
    if (n > 1) ans = ans / n * (n - 1);
    return ans;
}
```

1.2.2 数论函数的基本形状

1.3 同余方程

1.3.1 同余基本理论

定理1.3.1 欧拉定理

若$(a,n) = 1$，则$a^{\varphi(n)} \equiv 1 (\mathrm{mod}\, n)$.

1.3.2 一次同余方程

定理1.3.2 中国剩余定理(孙子定理)

令$m_1, m_2, \ldots, m_n$是两两互质的整数[a]，$m = \prod_{i=1}^{n} m_i, M_i = m/m_i$，则对于任意的$n$个整数$a_1, a_2, \ldots, a_n$，对于方程组(7):

$$\begin{cases} x \equiv a_1 (\mathrm{mod}\, m_1) \\ x \equiv a_2 (\mathrm{mod}\, m_2) \\ x \equiv a_3 (\mathrm{mod}\, m_3) \\ \ldots \\ x \equiv a_n (\mathrm{mod}\, m_n) \end{cases} \tag{7}$$

有整数解$x = \sum_{i=1}^{n} a_i M_i M_i^{-1}$，其中$M_i M_i^{-1} \equiv 1 (\mathrm{mod}\, m_i)$.

[a]这个性质使得在实际应用中很少用得上，更多的是递推用扩展欧几里得算法求解。

下面是示例代码[3][4]，注意b换成m即为上面所给定理的字母。

[2]该性质可以用于给欧拉函数打表。

[3]练习题：https://www.luogu.com.cn/problem/P3868

[4]这个代码的suml函数是龟速乘，其作用为防止在模运算中，$a * b$的结果爆整数上限。$smul(a,b,m) = ab \bmod m$

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
ll a[13], b[13];
ll k, B = 1;
ll exgcd(ll a, ll b, ll& x, ll& y){
    if (b == 0){
        x = 1;
        y = 0;
        return a;
    }
    ll ans = exgcd(b, a % b, x, y);
    ll nx = y, ny = x - (a / b) * y;
    x = nx;
    y = ny;
    return ans;
}
ll inv(ll a, ll m){
    ll x, y;
    exgcd(a, m, x, y);
    ll ans = (x % m + m) % m;
    return ans;
}
ll smul(ll a, ll b, ll m){
    ll ans = 0;
    while(b){
        if (b & 1){
            ans = (ans + a) % m;
        }
        b = b >> 1;
        a = (a + a) % m;
    }
    return ans;
}
ll CRT(){
    ll ans = 0;
    for (int i = 1; i <= k;i++){
        ll B_i = B / b[i];
        ans += smul(smul(a[i], B_i, B), inv(B_i, b[i]), B);
    }
    return (ans % B + B) % B;
}
int main(){
    cin >> k;
    for (int i = 1; i <= k;i++)
        cin >> a[i];
    for (int i = 1; i <= k;i++){
        cin >> b[i];
        B *= b[i];
    }
    cout << CRT() << endl;
}
```

1.3.3 高次同余方程

警告

阅读本节前，请先阅读**指数、原根与指标**一节

1.4 指数、原根与指标

2 抽象代数初步

> 噢，你是知道作者可怜的小脑袋瓜是理解不到抽象代数的一大堆抽象概念的，它是真抽象。——佚名

内容提要

-