ACM Template

目录

[ACM Template 1](#_Toc142920639)

[数学 1](#_Toc142920640)

[欧拉函数 1](#_Toc142920641)

# 数学

## 实用数学结论

逆元存在的条件

在模p的群中 则称x为a的逆(inv).

逆元存在的条件是

鸽笼原理 Pigeonhole Principle：

13个人中至少2个人的生日在同一个月

找规律

通过通向表达式找规律，将数字用a，b等字母替代即可

中位数

中位数的性质：所有数与中位数的绝对差之和最小

## 欧拉函数

中与 互质的数的个数，被称为欧拉函数

// 1~N 中与 N 互质的数的个数，被称为欧拉函数

//时间复杂度为根号n

inline int euler\_one(int n) {

int ans = n;

for (int i = 2; i \* i <= n; ++i) {

if (n % i == 0) {

ans = ans / i \* (i - 1);

while (n % i == 0)

n /= i;

}

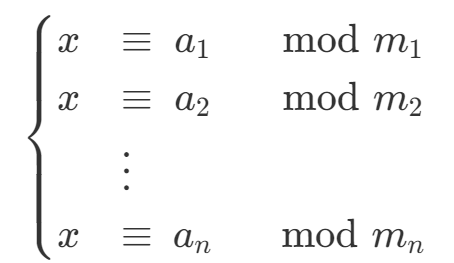
}

if (n > 1) ans = ans / n \* (n - 1); // n至多有一个比根号n大的质因子

return ans;

}

## 中国剩余定理：韩信点兵



1.若两两互质则有解

2. 在，下有唯一解为

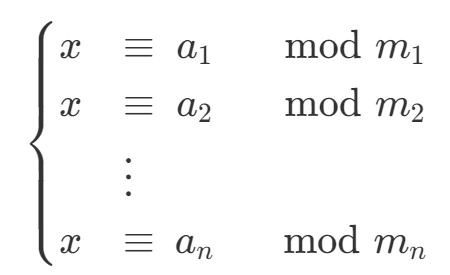
并设 是除了 以外的 个整数的乘积。

设 为 模 意义下的逆元。

理解性解释：

中国剩余定理主要用于解决这样的线性同余方程组，其中

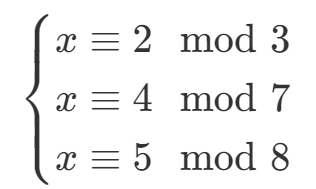
两两互质



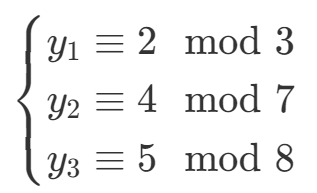
对于这样的方程组我们只需要求出 的一组特解，把这个特解加上 的最小公倍数的任意整数倍就可以得到所有解，所以我们就只需要考虑怎么构造出一组特解

例子

下面举个例子来帮助理解



直接构造不太行，但是对于每一个线性同余方程构造出一个特解是非常轻松的，所以我们先设第 个式子的解是 ，那么有



最终的 一定是 以各种方式组合起来，那么问题就是怎么组合才能得到

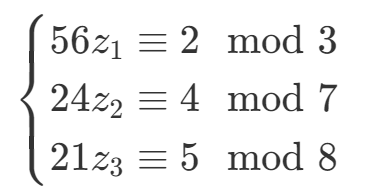
，如果 之间相乘就把问题变得更复杂了，所以我们把它们加起来，想办法凑出 如果 ，那么每一个 需要满足什么条件呢？因为 ，所以 一定是 3 的倍数，当 分别都是 3 的倍数时这一定也是满足条件的一个特解

所以我们不妨设 为3的倍数

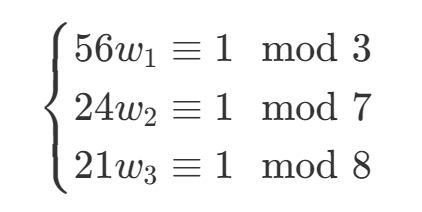
同理 为7的倍数，为8的倍数

那我们就可以把 写成 ，把 写成 ，把 写成

原方程就变成了这样：



然后就非常好求了，只需要求出



也就是 56,24,21 分别对于 3,7,8 的乘法逆元，然后在等式两边同时分别乘以 2,4,5 就可以得到 的一组特解,用扩展欧几里得算法求出 ，所以 于是

最后就得到 由于3,7,8的最小公倍数是168，我们将1229对于168取模就能得到 的最小正整数解

的通解就能表示为

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

#define int long long

const int N = 15;

int n, a[N], mods[N];

int exgcd(int a, int b, int& x, int& y) {

if (!b) {

x = 1, y = 0;

return a;

} else {

int gcd = exgcd(b, a % b, y, x);

y -= x \* (a / b); // 此处一定要先除后乘不然可能爆精度建议加括号

return gcd;

}

}

inline ll inv(ll x, ll p) // 求逆元

{

ll X, Y;

exgcd(x, p, X, Y);

return (X % p + p) % p;

}

inline ll crt() // CRT

{

ll prod = 1, ans = 0;

for (int i = 1; i <= n; ++i)

prod \*= mods[i];

ll tmp;

for (int i = 1; i <= n; ++i) {

tmp = prod / mods[i];

ans += (inv(tmp, mods[i]) \* a[i] \* tmp) % prod;

}

return ans % prod;

}

/\*

a[i]为每个方程式的余数

mods[i]为每个方程的模数

使用条件为所有mods[i]互质

\*/

signed example() {

cin >> n;

for (int i = 1; i <= n; ++i)

cin >> a[i] >> mods[i];

int ans = crt();

cout << ans;

}

## 数论基本知识

数论函数：定义域在整数上的函数

符号说明：(a,b)=c 相当于gcd(a,b)=c

积性函数：时

积性函数的性质： 则的全展开

积性函数f被所有处的值决定的

完全积性函数:不要求(a,b)=1即可f(ab)=f(a)f(b)

典型函数代表：

f(x)=1 I(x)=1

f(x)=x id(x)=x

f(x)=1 (x==1) =0(x!=1) e(x)=1 (x==1) =0(x!=1)

[x=1] 表示x==1时表达式=1，否则=0