数论基础

lizehon

Yali High School

2019 年 8 月 6 日

- 由于本人水平有限，不免出现纰漏，请各位大佬多多指正。
- 由于时间有限，部分大家较熟悉的部分将较少涉及。
- 选题较水，请放心食用。
- 欢迎大家积极参与讨论。

主要内容

- 扩展欧几里得
- 中国剩余定理
- 扩展中国剩余定理

主要内容

- BSGS
- 卢卡斯定理
- 扩展卢卡斯
- miller-rabin 筛素数
- pollard-rho 质因数分解
- ...

欧几里得算法

相信大家都会。
当 b 不为 0 时，$gcd(a, b) = gcd(b, a\%b)$

一般 gcd 都这样写：

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

有关定理及证明

- 定理 1

  设 a 和 b 不全为 0, 则存在整数 x 和 y, 使得 $ax + by = gcd(a, b)$

有关定理及证明

- 证明

在欧几里得算法的最后一步，当 $b=0$ 时,$gcd(a,b)=a$. 此时有一组解为 $x=1,y=0$
当 $b!=0$ 时，我们递归求 $gcd(b,a\%b)$. 假设存在一组整数 x',y' 满足 $bx'+(a\%b)y'=gcd(b,a\%b)=gcd(a,b)$.
那么,$bx'+(a-a/b\times b)y'=gcd(a,b)$ (/为整除)
所以,$ay'+b(x'-(a/b)y')=gcd(a,b)$
所以，令 $x=y',y=x'-(a/b)y'$，就得到了 $ax+by=gcd(a,b)$.
对欧几里得算法的递归过程运用数学归纳法证明，可知定理 1 成立.

扩展欧几里得算法

我们可以用以下代码求出 $ax + by = gcd(a, b)$ 的一组整数解 (x, y):

```
inline void Exgcd(int a, int b, int& d, int& x, int& y){
    if(b == 0){
        d = a, x = 1, y = 0;
        return;
    }
    Exgcd(b, a % b, d, x, y);
    int t = x; x = y, y = t - (a / b) * y;
}
```

其中 $d = gcd(a, b)$

有关定理及证明

- **定理** 2

  对于不定方程 $ax + by = c$, 当且仅当 $gcd(a, b)|c$ 时, 方程有整数解.

有关定理及证明

- **证明**

当 $gcd(a,b)|c$ 时，设 $g=gcd(a,b)$，$a'=a/g$，$b'=b/g$，$c'=c/g$.
用 $Exgcd(a',b')$ 求出不定方程 $a'x'+b'y'=1$ 的整数解 (x',y').
那么 $a'c'x'+b'c'y'=c'$
$ac'x'+bc'y'=c$
所以，$x_0=c'x',y_0=c'y'$ 是方程的一组解.
原方程 $ax+by=c$ 等价于 $a'x+b'y=c'$，所以通解为 $x=x_0+b'k,y=y_0-a'k,k\in Z$.
当 $gcd(a,b)$ 不整除 c 时，就没有上述求解过程，所以方程无解.

模板题

[LOJ10209] 青蛙的约会
link

两只青蛙在一个长度为 L 格的环上朝同一个方向跳。A 青蛙初始在 x，每次跳 m 格。B 青蛙初始在 y，每次跳 n 格。求至少跳几次使得两只青蛙在同一个点上。如果不可能碰面，输出 $Impossible$。
$0 < x, y, n, m, L \leq 2 \times 10^9$

模板题

设两只青蛙跳了 T 步，则 A 的坐标为 $x+mT$，B 的坐标为 $y+nT$。他们相遇的条件为 $x+mT-(y+nT)=LP(P\in Z)$。
即 $(n-m)T+LP=x-y, L>0$

模板题

$Exgcd(n-m, L, d, x, y)$ 得到一个解 x。
如果 $(x-y)\%gcd(n-m, L)! = 0$ 或者 $m=n$ 则无解。
设 $d=gcd(n-m, L)$，特解为 $x=x*(x-y)/d$，
通解为 $x=x*(x-y)/d+k(L/d)$。
最小整数解就很好求了，为 $(x\%(L/d)+L/d)\%(L/d)$。

补充内容 1：乘法逆元

后面内容的基础，大家都会。

费马小定理：$a^{p-1} \equiv 1 \pmod{p}$
当模数 p 是质数时，计算 $\frac{1}{a}$ 对 p 取模等价于 a^{p-2}。

中国剩余定理 CRT

- **应用范围**

当 $m_1 \dots m_n$ 两两互质时，求以下关于 x 的方程组的整数解

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

中国剩余定理 CRT

- 结论

令 $M=\prod m_i, M_i=M/m_i$,
t_i 是线性同余方程 $M_it_i\equiv 1 \pmod{m_i}$ 的一个解。
则 $x=\sum a_iM_it_i$。
且 x 在模 M 意义下有唯一解。

中国剩余定理 CRT

- **证明**

  因为 $M_i = M/m_i$ **是除** m_i **之外所有模数的倍数，所以**
  $\forall k \neq i, a_iM_it_i \equiv 0 \pmod{m_k}$
  所以 $x = \sum a_iM_it_i \equiv a_iM_it_i \equiv a_i \pmod{m_i}$。

中国剩余定理 CRT

- 应用

  中国剩余定理给出了模数两两互质的线性同余方程的一个特殊解。方程的通解可以表示为 $x + kM, k \in Z$。
  有些题目要求最小的非负整数解，因为 x 在模 M 意义下有唯一解，只需要把 x 对 M 取模，让 x 落在 $[0, M-1]$ 的范围内即可。

模板题

[LOJ10209] 曹冲养猪

link

给定 n 组限制，每组限制为：如果建了 a_i 个猪圈，则有 b_i 头猪没有去处。保证 a_i 互质。
求至少养了多少头猪。
$n \leq 10$ (范围太小了吧)

模板题

中国剩余定理模板题目，注意细节。

扩展中国剩余定理 EXCRT

当 m_i 不满足两两互质时，也是可以做的。这就是扩展中国剩余定理 $(EXCRT)$

模板题：[POJ2891]Strange Way to Express Integers
link
题意和模板题一样，但是不保证 m_i 两两互质。
没有数据范围 . . .

本题不保证 m_i 两两互质，下面给出 $EXCRT$ 算法。

可以考虑数学归纳法，假设已经求出了前 $k-1$ 个方程构成的方程组的一个解 x，记 $M = lcm(m_1 \dots m_{k-1})$，则 $x + iM(i \in Z)$ 是前 $k-1$ 个方程的通解。
考虑第 k 个方程，求出一个整数 t，使得
$x + t * M \equiv a_k \pmod{m_k}$。
该方程等价于 $M * t \equiv a_k - x \pmod{m_k}$，其中 t 是未知量。即这是一个线性同余方程，用 $Exgcd$ 判断是否有解，并求出它的解。
若有解，$x' = x + t * M$ 就是前 k 个方程构成的方程组的解。

综上所述，做 n 次 $Exgcd$ 即可。

Baby step giant step

- 应用范围

  给定整数 a, b, p，其中 a, p 互质，求最小的非负整数，使得 $a^x \equiv b \pmod p$。
  看名字就知道，这个算法就是分块，具体来说就是分成 $\sqrt{p}$ 块再 $O(\sqrt{p})$ 判断。

Baby step giant step

- 做法

设 $x = im - j$，其中 $m = \lceil \sqrt{p} \rceil$，$0 \leq j \leq m-1$。
则方程变为 $a^{im-j} \equiv b \pmod{p}$。
即 $(a^m)^i \equiv b \times a^j \pmod{p}$。
把右边所有的取值放入 map，在 map 中查找是否存在左边的取值。
做完了。

[SDOI2011] 计算器

[SDOI2011] 计算器
link

设计一个计算器完成一下三个任务:
1. 给定 y, z, p，计算 $y^z \mod p$ 的值。
2. 给定 y, z, p，计算满足 $x * y \equiv z \pmod p$ 的最小非负整数 x。
3. 给定 y, z, p，计算满足 $y^z \equiv z \pmod p$ 的最小非负整数 x。
或者判断无解。
多组数据，$1 \le T \le 10; 1 \le y, z, p \le 10^9$，保证 p 为质数。

[SDOI2011] 计算器

三个模板放在一起了。山东省果然是传统的数学大省。

补充内容 2：二项式定理

大家都会的东西。

$(a+b)^n = \sum_{k=0}^{n} C_n^k a^k b^{n-k}$

Lucas 定理

- **应用范围**

  当 p **为质数时，对于任意整数** $1 \le m \le n$**，有**
  $C_n^m = C_{n \bmod p}^{m \bmod p} \times C_{n/p}^{m/p} \pmod p$

Lucas 定理的证明

- **引理 1**

  由费马小定理得 $x^p \equiv x \pmod p$
  $(x+1)^p \equiv x+1 \equiv x^p+1 \pmod p$

- **引理 2 二项式定理**

  即 $(x+1)^n = \sum_{i=0}^{n} C_n^i x^i$

Lucas 定理的证明

- **证明**
  以下所有运算都在模 p 意义下进行，形如 (n/p) 的式子表示向下取整

$$(x+1)^n = (x+1)^{(n/p)p} \times (x+1)^{n\%p}$$

$$(x+1)^n = (x^p+1)^{(n/p)} \times (x+1)^{n\%p}$$

Lucas 定理的证明

二项式定理展开得到：

$\sum_{i=0}^{n} C_n^i x^i = (\sum_{i=0}^{(n/p)} C_{(n/p)}^i x^{pi})(\sum_{i=0}^{n\%p} C_{(n\%p)}^i x^i)$

当等式左边 $i = m$ 时，等式右边能组合出 x^m 的就是 $x^{(m/p)p}$ 和 $x^{m\%p}$

此时右边的 $i_1 = (m/p)$，得到 $C_{(n/p)}^{(m/p)} x^{(m/p)p}$；

$i_2 = m\%p$，得到 $C_{n\%p}^{m\%p} x^i$。

这样左右两边的系数相等，卢卡斯定理得证。

扩展 Lucas

- 应用范围

  当 p 不是质数时，求 $C_n^m \mod p$。
  p 必须满足所有质因子指数都是 1 才能这样做，到下一页你就明白了。

扩展 Lucas 做法

- 做法

  把 p 分解质因数得到 $a_1 \dots a_k$ 共 k 个质因子，
  分别计算 C_n^m 对 a_i 取模的结果 (需要分别预处理 $[1, a_i]$ 的对 a_i 的逆元)，设得到结果为 $b_1 \dots b_k$。
  再用中国剩余定理求解线性同余方程组：

  $$\begin{cases} x \equiv b_1 \pmod{a_1} \\ \dots \\ x \equiv b_n \pmod{a_n} \end{cases}$$

  得到最终答案。(比较难码)

[SDOI2010] 古代猪文

[SDOI2010] 古代猪文 (猪国杀那一套毒瘤题)
link

求 $q^{\sum_{d|n} C_n^d} \mod 999911659$
$q, n \leq 10^9$

这是一道涵盖面较广的一道基础的数论题。

[SDOI2010] 古代猪文

不要忘记了：如果 $q = 999911659$，则答案为 0。

999911659 是质数，由费马小定理，关键是求指数 $\sum_{d|n} C_n^d$ 对 999911658 取模的值。

写个程序分解质因数，发现 $999911658 = 2 \times 3 \times 4769 \times 35617$，然后做扩展 $Lucas$ 就行了。

最后得到指数 x，则答案为 $q^x \mod 999911659$。

miller-rabin

- 应用范围

  比较快地判断一个大数是否为质数，不写高精，适用范围就是 long long.

- 引理 1　费马小定理
  设 p 是素数，a 为整数，且 $(a,p)=1$，则 $a^{p-1}\equiv 1 \pmod p$。
- 前面没证，这里还是证一下吧。
  考虑 $S_1=\{1,2,3\dots p-1\}$，给它们同时乘上 a，得到 $S_2=\{a,2a\dots(p-1)a\}$。
  S_1 里面的数对 p 取模取遍了 $[1,p-1]$；由于 $(a,p)=1$，S_2 里面的数对 p 取模也是互不相等的，即也取遍了 $[1,p-1]$。
  所以 $1*2*\dots*(p-1)\equiv a*2a*\dots*(p-1)a \pmod p$
  所以 $(p-1)!\equiv(p-1)!a^{p-1} \pmod p$
  又因为 $((p-1)!,p)=1$
  所以 $a^{p-1}\equiv 1 \pmod p$

- 引理 2　　二次探测定理

  p 是一个质数，且 $0 < x < p$。如果 $x^2 \equiv 1 \pmod p$，则 $x = 1$ 或 $p - 1$

- 证明

  实际上 $x \equiv \pm 1 \pmod p$，$p - 1$ 就是 $-1 \pmod p$

- **理清思路**

  假设 n 是奇素数，则令 $n-1=2^q*m$。
  随便选一个整数 $a(0<a<n)$，显然有 $(a^{2^q*m}=a^{n-1})\equiv 1 \pmod{n}$。
  所以 $a^{2^{q-1}*m}\equiv \pm 1 \pmod{n}$。实际上 $(-1)^2=1$，所以如果出现了 -1，一定是 $a^m\equiv -1 \pmod{n}$。
  对于当前的测试，如果不满足 $a^{2^{q-1}*m}\equiv 1 \pmod{n}$，则 n 不是质数；否则继续测试 $a^{2^{q-2}*m}\dots$ 直至 $a^{2m}\equiv 1 \pmod{n}$ 而且 $a^m\equiv \pm 1 \pmod{n}$，则 n 通过了 $miller$ 测试，n 有很大概率为质数。

- **正确性分析**

  可以证明 $Miller - Rabin$ **算法给出错误结果的概率** $\leq \frac{1}{4}$**。若反复测试** k **次，则错误概率降低至** $(\frac{1}{4})^k$**，一般测试** 20 **次(看心情)。**

- 补充说明

求一个 long long 数的平方要用到快速乘。这个东西和快速幂差不多，$a * b$ 就是把 b 个 a 相加，看一眼代码就懂了。

```
inline LL qmul(LL a, LL b, LL mod){ // (a*b)%mod
    LL res = 0;
    for(; b; b >>= 1, a = (a + a) % mod)
        if(b&1) res = (res + a) % mod;
    return res;
}
```

看完以后，我们还是叫它龟速乘吧。

Pollard rho

- 应用范围
  对一个大数质因数分解

  实质上一种十分优秀的随机算法

- **算法流程**

  对于一个大数 n, 直接 $O(\sqrt{n})$ 是肯定不行的，我们尝试通过一些特殊的手段找到 n 的一个约数 p。
  如果能找到一个 x，使得 $gcd(n,p)>1$，那么就找到了 n 的两个约数。然后递归找下去，配合 $miller-rabin$ 判断质数。
  关键是怎么找这个 x。
  $Pollard-rho$ 算法是构造一个数列 $a_i=f(a_{i-1})$，其中 $f(x)=x*x+c$，c 是自己随机的一个数，然后每次选 a_i-a_{i-1} 或者 a_i-a_1 作为 x。

- 为什么是 $f(x) = x * x + c$
如果纯随机两个数，据说论文里面证明了大约要选 $n^{\frac{1}{4}}$ 个数，不够优秀。用各种复杂的复数性质可以证明这个函数是一个比较优秀的随机数。(不会证啊)

- 注意事项
1. 实际上这是一个伪随机数，因为这个数列里面包含有模 n 意义下的" 死循环"。所以如果出现 a_i 和 a_1 相等了就要换一个 c 重新找。
2. 有一个倍增的优化，就是每次只找 2^k 个数，要是还没找到 p 而且没有死循环就把范围扩大 $k <<= 1$。

[NOI2018] 屠龙勇士

你要按顺序杀死 n 条巨龙，每条龙初始血量为 a_i，初始你有 m 把剑，每把剑攻击力为 b_i。每次选择一把当前拥有的，攻击力不高于巨龙初始生命值中攻击力最大的一把剑，攻击 x 次，且只能攻击这 x 次。
游戏有这样的设定：每次攻击后它不断回血直到血量非负。只有在攻击后某个时刻生命值为 0 它才会死。
杀死一条巨龙，你会失去使用的这把剑，然后获得一把新的剑，攻击力 c_i。
你面对每条巨龙设置的 x 都是相同的，求将 x 设置为多少才能用最少的攻击次数通关。无解输出 -1。
有用的数据范围：对于所有的数据，$a_i \le p_i$ 或 $p_i = 1$。
对于所有的测试点，$T \le 5$，所有武器的攻击力 $\le 10^6$，所有 p_i 的最小公倍数 $\le 10^{12}$。

[NOI2018] 屠龙勇士

首先我们可以发现杀第 i 条龙的剑的攻击力是一定的，$multiset$ 的 $upper\_bound$ 可以做。还有一个性质就是每一把必须先砍到负数或者 0，不然就赢不了。

发现回血这个设定很像取模。对于每一条龙，只有满足 $C_i x \equiv A_i \pmod{P_i}$ 它才会死。其中 C_i 是剑的攻击力，A_i 是初始血量，P_i 是第 i 条龙的 $buff$。

对于 $p_i = 1$ 的情况很好解决，答案就是让每条龙血量小于等于 0 的最小刀数取 max。

接下来讨论 $A_i \le P_i$ 的情况。

[NOI2018] 屠龙勇士

求解方程组 $C_i x \equiv A_i \pmod{P_i}$，且 $A_i \le P_i$。
$EXCRT$ 的标准形式要求 $C_i = 1$，而且 C_i 模 P_i 不一定存在逆元，不能直接除掉。

原方程可以写作 $C_i x + P_i y = A_i$
$Exgcd$ 得到一组特解 x_0, y_0，则 $x = x_0 + k\frac{P_i}{gcd(C_i, P_i)}$
两边同时对 $\frac{P_i}{gcd(C_i, P_i)}$ 取模，得到 $x \equiv x_0 \pmod{\frac{P_i}{gcd(C_i, P_i)}}$
至此，已经转化为 $EXCRT$ 的标准形式。

记得用龟速乘。

[NOI2018] 屠龙勇士

- 总结

  $EXCRT$ 的题都挺套路的，冷静地转化就可以了。这就跟姜兴说三角函数是最套路的东西一样，太套路了。
  所以其他题目我就不讲了。(而且题目本来就少)

「SHOI2015」超能粒子炮 · 改

输入 n, k，求 $\sum_{i=0}^{k} C_n^i \pmod{2333}$.

多组数据，$T \le 10^5; n, k \le 10^{18}$.

「SHOI2015」超能粒子炮 · 改

考虑分析这个问题，设 $p=2333$，那么可以写成

$$ans=\begin{cases}C_{n\%p}^{0}C_{n/p}^{0}+C_{n\%p}^{1}C_{n/p}^{0}+\cdots+C_{n\%p}^{p-1}C_{n/p}^{0}\\ \ldots\\ C_{n\%p}^{0}C_{n/p}^{(k/p)+1}+C_{n\%p}^{1}C_{n/p}^{(k/p)+1}+\cdots+C_{n\%p}^{p-1}C_{n/p}^{(k/p)+1}\end{cases}$$

$$+C_{n\%p}^{0}C_{n/p}^{k/p}+\cdots+C_{n\%p}^{k\%p}C_{n/p}^{k/p}$$

设 $S_n^k=\sum_{i=0}^{k}C_n^i$，则

$$ans=S_n^k=S_{n\%p}^{p-1}S_{n/p}^{(k/p)+1}+S_{n\%p}^{k\%p}C_{n/p}^{k/p}$$

「SHOI2015」超能粒子炮 · 改

可以预处理 $S[5000][5000]$ 和 $C[5000][5000]$(想开多大随便)，然后就得到了一个类似 $lucas$ 的递归求解的过程就可以了。

- 为什么这么做

  普通的情况都是用逆元，但是当某个数对模数没有逆元的时候，就必须想办法转化了。

Summary

数论基础这一块主要是基础知识 (其实是网上没几道题，也有可能是我没找到)。
实际上如果出了这样的题目 (NOI2018 屠龙勇士)，那就按照相应的套路仔细地分析，不要慌。
应该还有很多套路是我不会的，在这里我只是把最最基础的部分展示了出来，大家还可以在网上多看一些资料 (虽然并不好找)。
因为信息还是需要很多数学知识的。

祝你们好运。

End

Thanks