数论基础

一、排列与组合

抽屉原理

- 桌上有十个苹果，要把这十个苹果放到九个抽屉里，无论怎样放，我们会发现至少会有一个抽屉里面至少放两个苹果。
- 这一现象就是我们所说的“抽屉原理”。抽屉原理的一般含义为：“如果每个抽屉代表一个集合，每一个苹果就可以代表一个元素，假如有n+1个元素放到n集合中去，其中必定有一个集合里至少有两个元素。”抽屉原理有时也被称为鸽巢原理。它是组合数学中一个重要的原理。
- **抽屉原理（鸽巢原理）**：把n+1件东西放入n个抽屉，则至少有一个抽屉放了两件或两件以上的东西。从另一个角度说，把n-1件东西放入n个抽屉，则至少有一个抽屉是空的。

抽屉原理

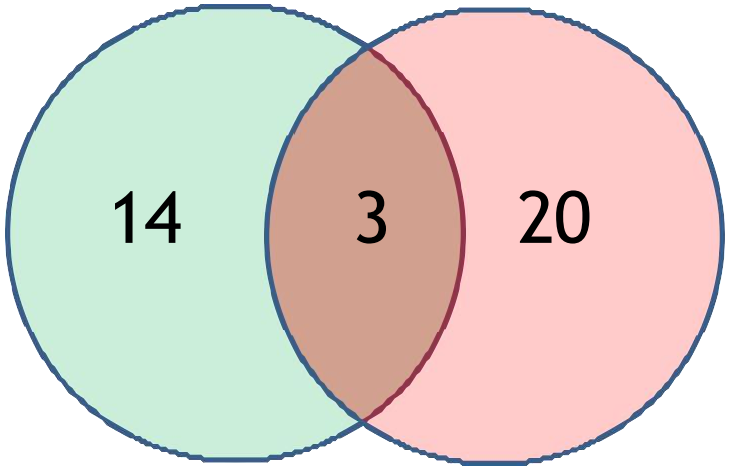
- 构造抽屉的方法：
- 运用抽屉原理的核心是分析清楚问题中，哪个是物件，哪个是抽屉。
- 例如，属相是有12个，那么任意37个人中，至少有几个人属相相同呢？这时将 属相看成12个抽屉，则一个抽屉中有 37/12，即3余1，余数不考虑，而向上考虑取整数，所以这里是3+1=4个人。
- 因此，在问题中，较多的一方就是物件，较少的一方就是抽屉，比如上述问题 中的属相12个，就是对应抽屉，37个人就是对应物件，因为37相对12多。

最差原则

- 最差原则，即考虑所有可能情况中，最不利于某件事情发生的情况。
- 例如，有300人到招聘会求职，其中软件设计有100人，市场营销有80人，财务管理有70人，人力资源管理有50人。那么至少有多少人找到工作才能保证一定有70人找的工作专业相同呢？
- 此时考虑最差情况：软件设计、市场营销和财务管理各录取69人，人力资源管理的50人全部录取，则此时再录取1人就能保证有70人找到的工作专业相同。 因此至少需要69*3+50+1=258人。

容斥原理

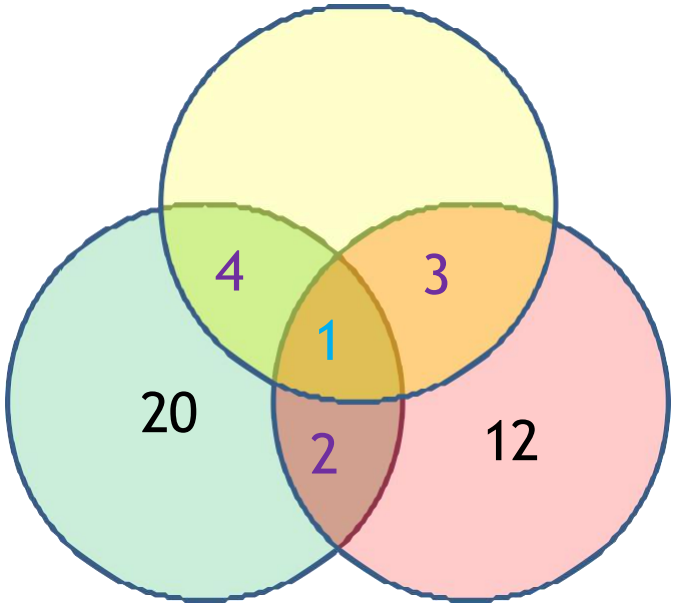
- 有14个同学学数学奥赛，20个同学学信息奥赛，3个同学两者都学。一共有多少同学学竞赛？



答案是14+20−3

容斥原理

- 有14个同学学数学奥赛，20个同学学信息奥赛，12个同学化学竞赛，4 个同学同时学数学和信息，3个同学同时学数学和化学，2个同学同时学信息和化学，1个同学三者都学。问一共有几个学竞赛的同学？



答案是14+20+12−4−3−2+1

加法原理

- 从甲地去乙地，可以乘火车，可以乘汽车，还可以乘轮船。一天中，火车有 4班，汽车有2班，轮船有3班，那一天中乘坐这些交通工具从甲到乙地有多 少种不同的选择？
- 在一天中，从甲地到乙地乘火车有4种选择，乘汽车有2种选择，乘轮船有 3种选择，以上无论选择了哪一种方法，都有可以从甲地到达乙地。因此 ，一天当中乘坐这些交通工具从甲地到乙地的不同选择共有4+2+3=9（种）。
- 把“从甲地到乙地”看成为“完成一件事” 完成它有三类方法（火车、汽车、轮船） 第一类有4种方法（火车有4班），第二类有2种方法（汽车有2班） ，第三类有3种方法（轮船有3班），因此完成一件事（从甲地到乙地）共有4+2+3=9种不同的方法。

加法原理

- 加法原理：做一件事，完成它有n类方法，第一类有m1种，第二类有m2 种，……，第n类有mn种，那么样完成这件事共有s=m1+m2+…+mn种方法。
- 加法原理的特点是：分类独立完成，分类计数。
- 书架上层有不同的数学书15本，中层有不同的语文书18本，下层有不同的物 理7本。现从其中任取一本书，问有多少种不同的取法？
- s=m1+m2+m3=15+18+7=40（种）

乘法原理

- 由A地去C地，中间必须经过B地，且已知由A 地到B地有3条路可走，再由B 地到C地有2条路可走，那么由A地经B地到C地有多少种不同的走法？
- 从A地到B地有3种不同的走法，分别用a1、a2、a3表示，而从B地到C地有2 种不同的走法，分别用b1、b2表示。所以从A地经B地到C地的全部走法有： a1b1；a1b2；a2b1；a2b2；a3b1；a3b2，共计6种。就是从A地到C地的3种走法与从C地到B地的2种走法的乘积，即 3×2=6（种）。
- 把“从A地到C地”看成完成一件事，完成这件事必须分二个步骤：第一个步骤有3种方法（从A地到B地）；第二个步骤有2种方法（从B地到C地）。因此“完成一件事”（从A地到C地）共有3×2=6（种）不同的方法。

乘法原理

- 乘法原理：做一件事，完成它需要n个先后步骤，做第一步有m1种不同的方法， 做第二步有m2种不同的方法，……，做第n步有mn种不同的方法，那么完成这件事共有s=m1×m2×…mn 种不同的方法。
- 乘法原理的特点是：分步依次完成，分步计数。
- 书架上层有不同的数学书15本，中层有不同的语文书18本，下层有不同的 物理7本，从中取出数学、语文、物理书各一本，问有多少种不同的取法？
- s=15×18×7=1890（种）

加法原理与乘法原理的联系与区别

- 加法原理与乘法原理的联系是：“完成一件事”。
- 区别是：
- 1. 加法原理：特点是分类独立完成。
- 2. 乘法原理：特点是分步依次完成。

排列

- 阶乘：n!=1*2*…*n（规定0!=1）
- 排列：有n个不同的物品，从中选出m个按顺序排成一列，问形成的排列的方案数。
- 对于第一个位置，我们有n种选择；第二个位置，有(n-1)种选择；……；第m个位置，有(n-m+1)种选择。Answer=n*(n-1)*…*(n-m+1)
- 我们将其记作：$A_n^m = \frac{n!}{(n-m)!}$
- 当m=n时，有A(n, n)=n!，即n的全排列；而把0<m<n的情况称为选排列。

组合

- 组合：有n个不同的物品，从中选出m个物品，不考虑顺序性，问方案数。
- 我们考虑n个物品选出m个的排列，由于在组合中不考虑顺序性，所以每一种组合在排列中重复出现了m!次，所以只需要将排列数除以m!。
- 我们将其记作：$C_n^m = \frac{n!}{(n-m)! * m!}$

排列与组合的区别

- 排列与组合的区别：取出的元素与顺序有关的为排列问题，与顺序无关的为组合问题。

例题

- 1. 用1 2 3 4 5这五个数字，组成没有重复数字的三位数，且为偶数，共有多少个？

- 2. 2名老师和4名学生排成一排，老师不站在两端，方案数有多少种？

- 3. 5名同学排成一排，甲与乙必须相邻的方案数有多少种？

例题

- 1. 用1 2 3 4 5这五个数字，组成没有重复数字的三位数，且为偶数，共有多少个？

$$2 * A_4^2 = 24$$

- 2. 2名老师和4名学生排成一排，老师不站在两端，方案数有多少种？（不相邻元素插空）

$$A_4^4 * C_3^2 * A_2^2 = 144$$

- 3. 5名同学排成一排，甲与乙必须相邻的方案数有多少种？（相邻元素捆绑）

$$A_4^4 * A_2^2 = 48$$

组合重要公式

$$C_n^m = C_n^{n-m}$$

$$C_n^m = C_{n-1}^m + C_{n-1}^{m-1}$$

杨辉等式

$$\sum_{i=0}^{n} C_n^i = 2^n$$

$$\sum_{i=0}^{n} (-1)\text{^}i * C_n^i = 0$$

奇数项和偶数项的和是相同的，都是2^(n-1)。

$$\sum_{i=0}^{n} (C_n^i)\text{^}2 = C_{2*n}^n$$

组合重要公式

- 这个性质非常重要 $C_n^m = C_{n-1}^m + C_{n-1}^{m-1}$
- 利用这条性质，我们可以在O(n^2)的时间的递推出所有0<=i<=n，0<=j<=i的C_i^j。
- 当j=0时，C=1。

```
for(int i=0;i<=n;i++)
{    C[i][0]=1;
     for(int j=1;j<=i;j++)
          C[i][j]=C[i-1][j]+C[i-1][j-1];
}
```

组合重要公式

$$(x+y)^n = \sum_{i=0}^{n} C_n^i x^i y^{n-i}$$

这个公式是二项式定理，可以证明第三个公式。

经典题

- 求有多少个长度为n的0/1序列，使得不存在两个1相邻。
- 答案对10^9+7取模。
- n<=100000

经典题

- 首先我们考虑枚举1的数量，然后把0排成一排。因为1之间是不能相邻的，所以就是用插空法求出一个组合数，累加起来即可。

[NOIP2016]组合数问题

- 给定k,n,m，有t组询问，每次询问给出x,y，问0<=i<=x，0<=j<=min(i,y)中有多少组(i,j)使得C_i^j是k的倍数。
- 保证x<=n,y<=m

- n,m<=2000，t<=10000

[NOIP2016]组合数问题

- 以下用C[i][j]表示C_i^j
- 由于n,m<=2000，所以我们可以在O(n^2)的时间内递推出所有的C[i][j]。

- 如果一个数a是k的倍数，那么a%k=0。
- 由于”+”与”%”可以混合运算，所以可以求出所有C[i][j]%k。
- 询问有多少个C[i][j]，是k的倍数，就是询问有多少个C[i][j]%k=0。

[NOIP2016]组合数问题

- 我们将C[i][j]%k填入一个以[0][0]为左上角、[n][m]为右下角的矩阵。
- 观察询问范围：0<=i<=x,0<=j<=min(i,y)
- 由于当j>i时C[i][j]无意义，那么每次询问的范围就是左下三角矩阵。
- 对C[i][j]%k=0的数量作二维前缀和，就可以每次O(1)回答询问。
- 时间复杂度$O(n^2+t)$。

[NOIP2011]计算系数

- 给定一个多项式$(ax+by)^k$，请求出多项式展开后$x^n y^m$ 项的系数，保证n+m=k
- 答案对10007取模

- k<=1000

[NOIP2011]计算系数

- 回归到题目，$(ax+by)^k$的$x^n y^m$项的系数就是
- C[n][k]*a^n*b^m
- 由于k<=1000，C[n][k]可以递推求得。

[NOIP2006]2^k进制数

- 题目：设r是个2^k 进制数，并满足以下条件：(1)r至少是个2位的2^k进制数。(2)作为2^k进制数，除最后一位外，r的每一位严格小于它右边相邻的那一位。(3)将r转换为2进制数q后，则q的总位数不超过w。1≤k≤9，k < w≤30000。
- 输入格式：k w
- 输出格式：一个十进制正整数，即满足条件的不同的的个数，要求r最高位不得为0。
- 输入样例#1：3 7
- 输出样例#1：36

[NOIP2006]2^k进制数

- 我们再从另一角度做一些解释：设s是长度为w 的01字符串（即字符串s由w个“0”或“1” 组成），s对应于上述条件(3)中的q。将s从右起划分为若干个长度为k 的段，每段对应一位2^k进制的数，如果s至少可分成2段，则 s所对应的二进制数又可以转换为上述的2^k 进制数r。
- 例：设k=3，w=7。则r是个八进制数（2^3=8）。由于w=7，长度为7的01字符串按3位一段分，可分为3段（即1，3，3，左边第一段只有一个二进制位），则满足条件的八进制数有：
- 2位数：高位为1：6个（即12，13，14，15，16，17），高位为2：5 个，…，高位为6：1个（即67）。共6+5+…+1=21个。
- 3位数：高位只能是1，第2位为2：5个（即123，124，125，126，127），第2位为3：4个，…，第2位为6：1个（即167）。共5+4+…+1=15个。
- 所以，满足要求的r共有36个。

[NOIP2006]2^k进制数

- 思路：组合公式+高精度
- 1. r最少为2位。
- 2. 这个数作为2^k进制数时，除了最后一位，每一位都严格小于右边一位。可以得出：构成这个数的元素互不相同，且一旦确定了选择哪些元素作为这个数的组成部分，那么排列的方式是唯一的，这暗示我们用组合数计算。
- 3. 可选的元素在1~2^k-1以内。
- 4. 一个w位的2进制数转化成2^k进制数，对应的最大位数为w/k，若不能整除，则为w/k+1。

[NOIP2006]2^k进制数

- 根据2^k进制时的位数，可以分为两种情况：
- 1. w%k=0，w位2进制数刚好凑成1个2^k进制数，位数i∈[2,w/k]，i位相当于是从1~2^k-1这些元素中选出i个。又根据上面的结论，一旦选出i个就只有一种排列方式，所以用组合数计算， 这种情况的个数：$\sum_{i=2}^{w/k} C_{2^k-1}^{i}$
- 注：这里讨论的是位数，位数不确定，但可选的元素确定！

[NOIP2006]2^k进制数

- 2. w%k!=0，可以确定至少要组成w/k位，然后剩下的w%k再构成多的一位。因此这种情况位数确定，但可选的元素要变化。讨论第w/k+1位的元素（最高位且不为0，因为为0时和w%k=0的情况相同）：
- a. 一旦这一位选了x，那么[1, x]区间的数都不能选。
- b. 在2进制的形式中，多出的这一位占有w%k位（2进制），因此可选的元素大小也就在 [1, 2^（w%k）-1]之间。
- c. 因为总共有2^k-1个元素，x及左端的不能选，因此每确定了最高位的元素x，就可以得 到剩下可选的元素个数为：2^k-1-x，这是变化的；而需要填的位数为w/k，这是不变的。

[NOIP2006]2^k进制数

- 所以这种情况个数：

$$\sum_{i=1}^{2^{w\%k}-1} C_{2^k-i-1}^{w/k}$$

- 这里i讨论的是可选的元素。
- 综上，结果是：

$$\sum_{i=2}^{w/k} C_{2^k-1}^{i} + \sum_{i=1}^{2^{w\%k}-1} C_{2^k-i-1}^{w/k}$$

- 因为组合数很大，需要用高精度。

二、基础数论算法

取整

- x是一个实数。
- floor(x) 对x向下取整。
- ceil(x) 对x向上取整。

- 在c++中，整型变量的除法都是整除的，我们一般考虑除数为正的情况。
- 若被除数为正，则向下取整；为负则向上取整。
- 总结为向绝对值小的方向取整。

整数除法的取整

$\lfloor\ \rfloor$下取整符号

$\lceil\ \rceil$上取整符号

c++中正整数的整除都是$\left\lfloor \frac{a}{b} \right\rfloor$向下取整。

整数除法的取整

$$\left\lfloor \frac{a}{b} \right\rfloor = \left\lceil \frac{a-b+1}{b} \right\rceil$$

$$\left\lceil \frac{a}{b} \right\rceil = \left\lfloor \frac{a+b-1}{b} \right\rfloor$$

扩展整除性质

$$a > \lfloor c/b \rfloor \Leftrightarrow ab > c$$

$$a < \lceil c/b \rceil \Leftrightarrow ab < c$$

$$a \le \lfloor c/b \rfloor \Leftrightarrow ab \le c$$

$$a \ge \lceil c/b \rceil \Leftrightarrow ab \ge c$$

可以自己来尝试证明一下。

取模

- a对b取模得到的结果就是a除以b的余数。
- 记作a mod b
- 例如10 mod 3 = 1

- x ≡ y (% p) 表示x与y对p取模的结果相等，称为同余。

取模

- 关于取模的几个基本性质：
- x+a ≡ y+a (% p)
- x-a ≡ y-a (% p)
- x*a ≡ y*a (% p) (以上假设x≡y (%p))
- (a + b)%p = (a%p + b%p)%p
- (a - b)%p = (a%p – b%p)%p
- (a - b)%p = (a - b + p)%p
- a*b%p = (a%p)*(b%p) %p
- 有了以上性质，我们就可以边计算边取模。

取模

- 在 c++ 中，注意
- 1. 一个正整数对一个正整数取模得到的是一个非负整数。
- 2. 一个负数对一个正整数取模得到的是负数或者是0。

$$a\%b = a-\left\lfloor \frac{a}{b} \right\rfloor *b$$

这是a为正，取模的定义式，很多时候会有妙用。

带余除法

- 对于两个正整数a，b，存在两个唯一的整数q，r满足

- b= a * q + r (0<=r<a)
- 其中b 对 a取模得到的是r
- 而b整除a为q

整除

- 整数a，b，且a不等于0

- 如果存在整数k满足b==a*k
- 那么a整除b，我们符号记做 a|b

- b称为a的倍数，a称为b的因数（或约数）

最大公约数

- 如果a%x = 0，我们称x是a的约数(或因数)，也称a是x的倍数。
- a与b的最大公约数，是指一个最大的整数x，使得x同时是a和b的约数。
- 我们将a与b的最大公约数记作gcd(a,b)。
- 例如：gcd(18,24)=6

- 那如何求解最大公约数呢？

最大公约数

- 欧几里得算法又称辗转相除法。
- 算法公式：

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

- 有了这个公式，我们就可以轻松求解最大公约数了，时间复杂度为log级。

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

最大公约数

- 首先为什么是log次，时间复杂度证明？
- 正确性？

最大公约数

- 我们考虑a 对 b取模，只要a大于b，那么a至少会减少一半，并且可能会减少更多。

- 也就是说每递归一次一个数会减少至少一半，每次除以2，所以就是log级别的了。

最大公约数

- 为什么gcd(a,b)=gcd(b,a%b) ？？？
- 设gcd(a,b)=d，a=md，b=nd
- 则gcd(m,n)=1(也称m与n互质)
- a%b=

$$a - \left\lfloor \frac{a}{b} \right\rfloor * b = md - \left\lfloor \frac{md}{nd} \right\rfloor * nd$$

$$= \left(m - \left\lfloor \frac{m}{n} \right\rfloor * n \right) d$$

$$= m\%n * d$$

- 所以gcd(b,a%b)=gcd(nd,m%n*d)=d*gcd(n,m%n)
- 因此只要证gcd(n,m%n)=1

最大公约数

- 假设gcd(n,m%n)=q≠1
- 设m=kn+r (0<=r<n)，则m%n=r，gcd(n,r)=q
- 设n=n’q，r=r’q
- 那么m=kn’q+r’q=(kn’+r’)q
- 那么q就成为m与n的公约数，这与我们所假设的gcd(m,n)=1矛盾
- 所以gcd(n,m%n)=1
- 所以gcd(b,a%b) = d*gcd(n,m%n) = d = gcd(a,b)
- 于是我们证明了该算法，可以放心使用啦！

最大公约数的性质

- 关于gcd还有以下扩展：
- gcd(a,b,c)=gcd(gcd(a,b),c)

- 将a与b的最小公倍数记作lcm(a,b)
- 则lcm(a,b)=a*b/gcd(a,b)
- (后面再作证明)

- 与gcd同理，lcm(a,b,c)=lcm(lcm(a,b),c)
- 但是lcm(a,b,c)=a*b*c/gcd(a,b,c)不成立！！！

扩展欧几里得算法

- 扩展欧几里得算法用于解决这样一个问题：
- 给定正整数a,b，求ax+by=gcd(a,b)的一组整数解。

- 考虑像欧几里得算法一样递归
- gcd(a,b)=gcd(b,a%b)
- 假设已知bx'+a%b*y'=gcd(b,a%b)的一组解x',y'
- 推出ax+by=gcd(a,b)的一组解x,y

扩展欧几里得算法

- ax+by=gcd(a,b)=bx'+a%b*y'
- $= bx' + (a - \left\lfloor \frac{a}{b} \right\rfloor * b)y'$
- $= ay' + b(x' - \left\lfloor \frac{a}{b} \right\rfloor * y')$
- 于是得到：$x = y';\quad y = x' - \lfloor a / b \rfloor * y'$
- 当b=0时，a*1+b*0=a=gcd(a,b)
- 这样就可以递归求解了。

扩展欧几里得算法

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

```
exgcd(b,a%b,x,y);
int t;
t=x;
x=y;
y=t-a/b*y;
```

扩展欧几里得

- 但是这只是ax+by=gcd(a,b)任意的一组整数解
- 如果要求ax+by=c呢？如果要求x>0且最小呢？
- 我们假设已经得到ax'+by'=gcd(a,b)的一组解
- 若求解ax+by=c
- 当c % gcd(a,b) ≠ 0时，无解
- 否则，令k=c/gcd(a,b)
- ax'k+by'k=gcd(a,b)*k=c
- 得到x=x'k，y=y'k

扩展欧几里得

- 如果要求x非负且最小呢？
- 我们假设已经得到ax+by=c的一组解
- 令g=gcd(a,b)
- lcm(a,b)=a*b/g
- ax+lcm(a,b)+by-lcm(a,b)=c
- a(x+b/g)+b(y-a/g)=c
- 由此可得，x加上或减去任意倍数的b/gcd(a,b)后均有对应的y的解
- 令t=b/gcd(a,b)，(x%t+t)%t就是x的最小非负解

[BZOJ1441]Min

- 给定一个长度为n的正整数序列A1…An，另有一个未知的整数序列X1…Xn
- 令S=A1*X1+A2*X2+…+An*Xn，请你确定一个X序列，使得S>0且S尽量小，只需输出最小的S值即可

- n<=100000，Ai<=10^9

[BZOJ1441]Min

- 我们考虑A数列只有两个数字a,b的情况
- ax+by=S
- 根据exgcd的知识，该方程有解当且仅当S是gcd(a,b)的倍数
- 换言之，此时S的最小值就是gcd(a,b)
- 推广到n>2的情况，与n=2时的证明方法类似，此时有解当且仅当S是gcd(A1,A2,...,An)的倍数，因此S_{min}= gcd(A1,A2,...,An)
- 时间复杂度O(n log 10^9)

同余方程

$$a * x \equiv c \pmod{p}$$

$$a * x - y * p = c$$

$$a * x + y * p = c$$

然后这就是不定方程的标准形式了

同时由此我们也可以知道若gcd(a,p)不能整除c，那么方程无解。

同余方程的一个性质

- 如果一个同余方程有解，那么在模意义下，它必定有恰好gcd(a,p)个解。

- 这个不是说很好严谨的证明，直观理解方便。

[NOIP2012]同余方程

- 给定正整数a,b，求ax ≡ 1 (% b)的最小正整数解，保证有解。
- a,b均在int范围内。

[NOIP2012]同余方程

- 这就是刚刚那个问题的特殊模型。
- $ax \equiv 1 \ (\% \ b)$
- $ax = 1+by$
- $ax-by = 1$
- 典型的exgcd的形式，求x的最小正整数解。
- 由于保证有解，相当于保证了gcd(a,b)=1。

扩欧经典必做题

- [青蛙的约会] luogu P1516
- 题意：有一个环形棋盘，每个格子编号为1..n，有两只青蛙A和B，起始坐标分别为x,y，每一秒，A向后跳a格，B向后跳b格，求最早几秒后两青蛙相遇或永远不会相遇。

- n,x,y,a,b均在int范围内。

扩欧经典必做题

- [青蛙的约会]
- 假设t秒后相遇，由题意得：
- x+at ≡ y+bt (% n)
- (a-b)t ≡ y-x (%n) （这就转化为同余方程了）
- (a-b)t + kn = y-x

- 然后就可以用exgcd求t的最小非负整数解了。

一道经典例题

- 求直线ax+by+c=0上有多少个整点(x,y)满足 $x\in[x_1,x_2]$ ， $y\in[y_1,y_2]$ 。

- 各参数都在int范围内。

一道经典例题

- 首先方程变成ax+by=-c，先看看-c是不是gcd(a,b)的倍数。然后用扩展欧几里得求一下ax+by=gcd(a,b)的解(x0,y0)。
- 等式两边同乘-c/gcd(a,b)，得到
- ax0’+by0’=-c
- 用刚才的结论求出使x=x0’+kb/gcd(a,b)落在区间[x_1,x_2]里的k的范围，出使y=y0’-ka/gcd(a,b)落在区间[y_1,y_2]里的k的范围，取交集即可。

质数

- 质数（素数）的定义：一个大于1的自然数，除了1和它自身外，不能被其他自然数整除的数叫做质数；否则称为合数。

质数的性质

- 1. 质数的个数是无穷的。（用欧几里得来证明）
- 2. 1..n质数个数记为 $\pi(n)$ ，其中估计值大约是 $\frac{n}{\log(n)}$
- 3. 如果一个数n(n>1)是一个合数，那么它必定会存在一个小于等于根号n的质因子。
- 4. 任意数最多只会有一个大于根号n的质因子。

质数的判定

- 1. 根据定义，枚举比它小的数判断，复杂度O(n)
- 2. 根据性质3，只需要枚举到根号n，复杂度O(n^(1/2))
- 3. 我们可以预处理根号n以内的质数，然后再利用性质3。预处理复杂度是筛法的复杂度。查询复杂度是O(n^(1/2)/log(n))。

唯一分解定理

- 唯一分解定理（也称基本算数定理）：任意一个正整数c，将其分解为若干质数的正整数次幂的乘积，该分解方法唯一。
- 形如：$c=p_1^{a1}*p_2^{a2}*\ldots\ldots*p_n^{an}$，$p_1 \ldots p_n$均为质数。

质因数分解

- 将正整数c，化作
- $c=p_1^{a1}*p_2^{a2}*\ldots\ldots*p_n^{an}$，($p_1 \ldots p_n$均为质数)的形式，这一过程叫做质因数分解。
- 质因数分解有显然的O(c)的做法，不再赘述，我们考虑更快的做法。
- 假如c有大于 $\sqrt{c}$ 的质因数，那么它仅有一个该类因数且次数为1。（性质4）

质因数分解

- 于是我们可以只枚举小于等于$\sqrt{c}$的数并判定其是否为c的因数，若是则从中除去。
- 若结束后，c仍大于1，那么此时的c就是那个大于$\sqrt{c}$的质因数。时间复杂度O($\sqrt{c}$)。

```
t=0;
for(int i=2;i*i<=c;i++)
    if(c%i==0)
    {   p[++t]=i,a[t]=0;
        while(c%i==0)c=c/i,++a[t];
    }
if(c>1)p[++t]=c,a[t]=1;
```

质因数分解与gcd,lcm

- 将两个正整数A,B质因数分解
- $A=p_1^{a1}*p_2^{a2}*\ldots\ldots*p_n^{an}$
- $B=p_1^{b1}*p_2^{b2}*\ldots\ldots*p_n^{bn}$
- 那么：
- $gcd(a,b)=p_1^{min(a1,b1)}*p_2^{min(a2,b2)}*\ldots\ldots*p_n^{min(an,bn)}$
- $lcm(a,b)=p_1^{max(a1,b1)}*p_2^{max(a2,b2)}*\ldots\ldots*p_n^{max(an,bn)}$
- 由于max(a,b)=a+b-min(a,b)
- 所以易证明lcm(a,b)=a*b/gcd(a,b)

阶乘的0

- 给定n，求n!末尾0的数量。

- n<=1000000

阶乘的0

- 唯一分解之后，2*5会形成一个10，也就是会在末尾形成一个0。

- 然后2的个数一定是多于5的数量的，所以我们问题就转化为了求n!里面5的数量。

质数筛法

- 如果我们想要求出n以内的所有质数，最简单的方法可以枚举每个数，判断是否为质数，时间复杂度为$O(n\sqrt{n})$，太慢！

- 对此我们有筛法。
- 常用的有两种：
- 埃氏筛法，时间复杂度O(nlog(logn))
- 欧拉筛法，俗称线性筛法，时间复杂度O(n)

埃氏筛法

- 将一个大小为n的数组a置为1，枚举n以内的每个数，将以其倍数为下标的位置置为0，最终a数组内为1的位置下标为质数，正确性由质数的定义可知。

```
for(int i=1;i<=n;i++)a[i]=1;
a[1]=0;
for(int i=2;i<=n;i++)
    for(int j=i*2;j<=n;j+=i)a[j]=0;
```

- 时间复杂度的证明需要使用调和级数：

$$1+\frac{1}{2}+\cdots+\frac{1}{n}\approx \ln(n+1)+0.577\cdots$$

欧拉筛法

- 观察埃氏筛法，其缺点在于一个位置可能被反复置0，浪费了时间，在欧拉筛法中，对于每个合数c，使得它只被作为其最小质约数的倍数筛掉。

```
for(int i=1;i<=n;i++)a[i]=1;
int t=0;
for(int i=2;i<=n;i++)
{    if(a[i]==1)p[++t]=i;
     for(int j=1;j<=t&&p[j]*i<=n;j++)
     {    a[p[j]*i]=0;
          if(i%p[j]==0)break;
     }
}
```

每个合数只被筛掉一次，复杂度O(n)

逆元

- 对于正整数a,b，如果能找到正整数x使得
- $ax \equiv 1 \ (\% \ b)$，我们称x是a在模b意义下的逆元。
- 在这里，实际上a与x互为模b意义下的逆元。
- 由同余方程可知，a在模b意义下存在逆元，当且仅当gcd(a,b)=1，即a与b互质。

- 逆元有什么用呢？

逆元

- 众所周知，在取模的意义下是不能直接作除法的。
- 例如：12%11=1，(12/3)%11=4
- 但是(1/3)%11≠4
- 但是，我们找到3在模11意义下的逆元4
- 发现(12*4)%11=4
- 逆元的作用：在模意义下，除以一个数，相当于乘上这个数的逆元。
- 这样我们就可以在模意义下作除法了！

求逆元

- 如何求解逆元呢？
- 首先，我们可以通过exgcd进行求解，该方法适用于所有有解的情况。
- 但是当模数为质数时，有另一种方法。

- **费马小定理**：$a^{p-1} \equiv 1 \ (\%\ p)$，p为质数，a与p互质

- 由此可得： $a*a^{p-2} \equiv 1 \ (\%\ p)$
- 所以，当模数为质数p时，a的逆元等于a^{p-2}。

费马小定理

- 其实在比赛当中很多的的计数题往往让你输出一个答案对一个数取模的结果，一般这个模数都是一个质数。

- 而是质数的话，用费马小定理就非常方便了。

快速幂

- 你说你不会求a^{p-2}？我们有O(logn)的快速幂！
- 为了求解a^n，我们将指数n划分为若干2的次幂的和。
- 例如，求解a^{26}，$26=2^1+2^3+2^4=2+8+16$
- 所以$a^{26}=a^2*a^8*a^{16}$
- 我们又发现$a^2=(a^1)^2$,$a^4=(a^2)^2$,$a^8=(a^4)^2$……
- 我们就可以用logn的时间求出a, a^2,a^4,a^8 ……
- 接下来只需要将需要的a的次幂相乘，即可得到答案。

快速幂

- 那么哪些次幂是需要的呢？

- 我们来观察指数的二进制。
- 例如，26的二进制为11010，从最低位起，0和1就分别表示了对应的次幂是否需要。

- 指数的二进制可以通过不断除2和模2来得到。

快速幂

```
int quick_power(int a,int b,int p)
{    int ans=1;
     while(b>0)
     {    if(b%2==1)ans=ans*a%p;
          a=a*a%p;
          b=b/2;
     }
     return ans;
}//求解a的b次方模p
```

[NOIP2013]转圈游戏

- n个小伙伴（编号从0到n-1）围坐一圈玩游戏。按照顺时针方向给n个位置编号，从0到n-1。最初，第0号小伙伴在第0号位置，第1号小伙伴在第1号位置，依此类推。游戏规则如下：每一轮第0号位置上的小伙伴顺时针走到第m号位置，第1号位置小伙伴走到第m+1号位置，依此类推，第n−m号位置上的小伙伴走到第0号位置，第n-m+1号位置上的小伙伴走到第1号位置，……，第n-1号位置上的小伙伴顺时针走到第m-1号位置。现在，一共进行了10^k轮，请问x号小伙伴最后走到了第几号位置。(n,m,k在int范围内)

[NOIP2013]转圈游戏

- 此题作为当年第一题，并不难。

- 位置为x的人，在一轮游戏后变为(x+m)%n
- 在r轮游戏后，变为(x+m*r)%n
- 题目要求进行10^k轮游戏，那么位置就变为
- (x+m*10^k)%n
- 直接求解即可，10^k用快速幂求解。

oi中可能用到的数列公式

$$\sum_{i=1}^{n} i = n*(n+1)/2$$

$$\sum_{i=1}^{n} i\text{^}2 = n*(n+1)*(2*n+1)/6$$

$$\sum_{i=1}^{n} i\text{^}3 = [\frac{n*(n+1)}{2}]\text{^}2$$

oi中可能用到的数列公式

等差数列求和公式 $\frac{(a[1]+a[n])*n}{2}$

等比数列求和公式 $\frac{a[1]*(1-q^n)}{1-q}$

三、概率和数学期望

概率

- 某个事件A发生的可能性的大小，称之为事件A的概率，记作P(A)。
- 假设某事的所有可能结果有n种，事件A涵盖其中的m种，那么P(A)=m/n。
- 例如投掷一枚骰子，点数小于3的概率为2/6=1/3。

概率

- 如果两个事件A和B所涵盖的结果没有交集（互不相容），那么P(A或B发生)=P(A)+P(B)
- 还是掷骰子
- P(点数小于3或点数大于4)=2/6+2/6=2/3

- 如果A和B所涵盖的结果有交集
- 那么P(A或B发生)=P(A)+P(B)-P(A与B同时发生)
- P(点数小于3或点数为偶数)=2/6+3/6-1/6=2/3

概率

- 记事件B为“事件A不发生”（事件A的对立事件，事件B发生则A不会发生）
- 那么P(A)+P(B)=1，即P(B)=1-P(A)
- P(点数不小于3)=1-2/6=2/3

- 在两个**互不干扰**的事中，事件A在其中一件事中，事件B在另外一件事中（独立事件，事件A发生跟B的发生没有关系）
- 那么P(A与B同时发生)=P(A)*P(B)
- 掷两个骰子
- P(第一个点数小于3且第二个点数为偶数)=(2/6)*(3/6)=1/6

数学期望

- 事件A有多种结果，记其结果为x，那么x的期望值表示事件A的结果的平均大小，记作E(x)
- E(x)=每种结果与其概率的乘积的和
- 例如，记掷一枚骰子的点数为x
- E(x)=1*(1/6)+2*(1/6)+3*(1/6)+4*(1/6)+5*(1/6)+6*(1/6)=7/2
- 若c为常数，那么：
- E(x+c)=E(x)+c，E(c*x)=c*E(x)

数学期望

- 记两个事件的结果分别为x,y
- E(x+y)=E(x)+E(y)
- 例如：E(语文成绩+数学成绩)=E(语文成绩)+E(数学成绩)

- 若两个事件互相独立，E(x*y)=E(x)*E(y)
- E(语文成绩*数学成绩)=E(语文成绩)*E(数学成绩)

[HNOI2008]越狱

- 监狱有连续编号为1...n的n个房间，每个房间关押一个犯人，有m种宗教，每个犯人可能信仰其中一种。如果相邻房间的犯人的宗教相同，就可能发生越狱，求有多少种状态可能发生越狱，答案对100003取模
- n，m在long long范围内

- 例如：当n=3,m=2时，答案为6

[HNOI2008]越狱

- 计数题有一种技巧叫做取补集
- 直接求可能发生越狱的方案数并不好求

- 可能发生越狱的方案数=
- 所有的方案数-不能发生越狱的方案数

- 所有的方案数=m^n
- 不能发生越狱的方案数=m*(m-1)^(n-1)
- Answer = m^n - m*(m-1)^(n-1)，快速幂即可

Thanks For Listening!