第九周直播课（下午）
数学与数论

洛谷网校
基础-提高衔接计划
2024-09
disangan233

洛谷
www.luogu.com.cn

目录

整除

- 整除：对两个正整数 a, b（$b \leq a$），如果存在一个整数 k，使得 $a = bk$，则称 b 整除 a，记作 $b \mid a$
- 不能在代码中写：`if (b | a)`（位运算或）
- 正确的写法：`if (a % b == 0)`

带余除法

- 带余除法：对任意整数 a 和正整数 b，一定存在一个整数 $r \in [0, b)$ 和一个整数 k，使得 $a = kb + r$。称上式为 a 和 b 的带余除法式，r 称为 a 除以 b 的余数
- 也就是我们小学数学所写的 $a \div b = k \cdots r$
- 求商：`k = a / b`
- 求余数：`r = a % b`
- C++11 标准起，规定商向零取整（舍弃小数部分），取模的符号与被除数相同：

```
5 % 3 == 2
5 % -3 == 2
-5 % 3 == -2
-5 % -3 == -2
```

因数

- 因数：对一个正整数 x，所有能整除 x 的正整数均为 x 的因数
- 公因数：两个正整数 x, y 共有的因数称为它们的公因数
- 质数：因数只有 1 和自身的正整数叫质数；1 不是质数
- 合数：大于 1 的正整数中，不是质数的数称为合数
- 质因子：x 的因数中，是质数的那部分被称为质因子

算术基本定理

- 算术基本引理：设 p 是质数，$p \mid a_1 a_2$，那么 $p \mid a_1$ 和 $p \mid a_2$ 至少有一个成立
- 算术基本引理是质数的本质属性，也是质数的真正定义，说明质数是不可拆分的
- 例如 $7 \mid (21 \times 4)$，有 $7 \mid 21$
- 合数是可拆分的，$4 \mid (2 \times 18)$，但是 4 既不整除 2 也不整除 18

算术基本定理

- 算术基本定理（唯一分解定理）：对任何一个大于 1 的整数 x，x 均能表示成若干个 x 的质因子的乘积。这个表示法是唯一的
- 例如，60 可以被唯一分解成 $2 \times 2 \times 3 \times 5$（不计次序）
- 合并相同质数，从小到大排序，称为标准质因数分解式：

$$n = p_1^{k_1} p_2^{k_2} \cdots p_s^{k_s}, \quad p_1 < p_2 < \cdots < p_s$$

- 例如，$60 = 2^2 \times 3 \times 5$

基础算法

- 因数分解：$O(\sqrt{x})$ 对任意正整数 x 分解因子

```
for (int i = 2; i * i <= x; i++) {
    if (x % i == 0)
        ans[++k] = i;
    if (i != x / i)
        ans[++k] = x / i;
}
sort(ans + 1, ans + k + 1);
```

基础算法

- 判断质数：$O(\sqrt{x})$ 判断 x 是否是质数

```
bool isprime(int x) {
    if (x <= 1)
        return 0;
    for (int i = 2; i * i <= x; i++)
        if (x % i == 0)
            return 0;
    return 1;
}
```

质数筛

- 如果我们想要知道小于等于 n 有多少个质数呢？
- 一个自然的想法是对于小于等于 n 的每个数进行一次质数检验，时间复杂度 $O(n\sqrt{n})$
- 这种暴力的做法显然不能达到最优复杂度
- 质数筛：更快地求出 $1 \sim n$ 内所有质数的算法

埃氏筛

- 考虑这样一件事情：对于任意一个大于 1 的正整数 n，那么它的 x 倍就是合数（$x > 1$）
- 利用这个结论，我们可以避免很多次不必要的检测
- 从小到大考虑每个数，然后同时把当前这个数的所有（比自己大的）倍数记为合数，那么运行结束的时候没有被标记的数就是质数了

埃氏筛

```
for (int i = 2; i <= n; i++) {
    if (vis[i]) continue; // 只用质数去筛
    prime[++k] = i;
    for (int j = 2 * i; j <= n; j += i)
        // 小优化：可以从 i * i 开始
        // 2 * i 到 (i - 1) * i 会被小于 i 的数先筛掉
        vis[j] = 1; // 是 i 的倍数的均不是质数
}
```

- 以上为埃拉托斯特尼筛法，简称为埃氏筛
- 时间复杂度为 $O(n \log \log n)$
- 注意 $\log \log n < \log n < \log^2 n$

埃氏筛

- 对于埃式筛的复杂度，不作具体证明
- 一些可以了解的内容：
- 质数分布：$\pi(n) \approx \frac{n}{\ln n}$，$\pi(n)$ 指小于等于 n 的质数个数
- 如果要求 10^8 内的质数，prime 数组大小就只用开 10^7
- 调和级数：$\sum_{i=1}^{n} \frac{n}{i} = n + \frac{n}{2} + \frac{n}{3} + \cdots + \frac{n}{n} < O(n \log n)$
- 对应以下代码的时间复杂度

```
1 for (int i = 2; i <= n; i++) {
2     for (int j = 2 * i; j <= n; j += i)
3         vis[j] = 1;
4 }
```

线性筛素数

- 如何线性筛出 n 以内的素数？
- 以下的代码可以保证每个数都被自己的最小质因子筛掉

```
void getprime() {
    for (int i = 2; i <= n; i++) {
        if (!vis[i])  p[++k] = i;
        for (int j = 1; j <= k && i * p[j] <= n; j++) {
            vis[i * p[j]] = 1;
            if (i % p[j] == 0)  break;
        }
    }
}
```

- 以上为欧拉筛，也叫线性筛

欧拉筛的证明

- 先证正确性：对合数 $x = py$，其中 p 是 x 的最小质因子
- 考虑当 $i = y$ 时，枚举已筛出的质数 j
- 由于 p 是最小质因子，i 不含小于 p 的质因子，所以在枚举到 p 前，$i \bmod j = 0$ 不会成立，循环不会终止
- 这就证明了 x 会被筛掉

欧拉筛的证明

- 再证明复杂度为线性：假设 $x = py = qz$，其中 p 是 x 的最小质因子，q 是另一质因子
- 根据唯一分解定理，z 一定含有质因子 p，且 $p < q$
- 则 $i = z, j = p$ 时，$i \bmod j = 0$ 会先一步成立，循环终止，质数不会枚举到 q
- 考察证明过程可以发现，j 就是合数 $i \times j$ 的最小质因子

快速质因子分解

- B3716 质因子分解 3
- T 组数据。每次给定一个整数 n，求 n 所有质因子的按位异或和
- $1 \leq T \leq 10^5$，$2 \leq n \leq 10^8$

快速质因子分解

- 如果每次把 n 都除以自己的最小质因子，一个数最多有 $O(\log n)$ 个质因子，那么可以 $O(\log n)$ 完成一次质因子分解
- 思考如何求出最小质因子？
- 在线性筛的时候，每个数都被自己的最小质因子筛掉。所以当一个数被筛的时候，记录筛掉它的数即可

快速质因子分解

```
1  void getprime() {
2      for (int i = 2; i <= n; i++) {
3          if (!vis[i])  p[++k] = i, mn[i] = i;
4          for (int j = 1; j <= k && i * p[j] <= n; j++) {
5              vis[i * p[j]] = 1;
6              mn[i * p[j]] = p[j];
7              if (i % p[j] == 0)  break;
8          }
9      }
10 }
11 int get(int n) {
12     int ans = 0;
13     while(n>1) ans^=mn[n],n/=mn[n];
14     return ans;
15 }
```

同余

- 对两个整数 a, b，如果它们除以 d 的余数相同，则称它们模 d 同余，记作

$$a \equiv b \pmod{d}$$

- 在同余号两侧同时加、减、乘任意整数，同余式仍成立
- a, b, r 的带余除法式的等价表达是 $a \equiv r \pmod{b}$
- 在模 d 同余下，只有 $0, 1, \ldots, d-1$ 这 d 个本质不同的数

最大公约数

- 最大公约数即为 Greatest Common Divisor，常缩写为 gcd
- 一组整数的公约数，是指同时是这组数中每一个数的约数的数
- 一组整数的最大公约数，是指所有公约数里面最大的一个
- 对不全为 0 的整数 a, b，将其最大公约数记为 $\gcd(a, b)$，不引起歧义时可简写为 (a, b)
- 对不全为 0 的整数 $a_1, \ldots, a_n$，将其最大公约数记为 $\gcd(a_1, \ldots, a_n)$，不引起歧义时可简写为 $(a_1, \ldots, a_n)$。

欧几里得算法

- 如果我们已知两个数 a 和 b，如何求出二者的最大公约数呢？
- 不妨设 $a > b$
- 如果 b 是 a 的约数，那么 b 就是二者的最大公约数
- 若 b 不能整除 a，有 $a = b \times k + c$，其中 $c < b$。此时可以通过证明得到 $\gcd(a, b) = \gcd(b, a \bmod b)$

欧几里得算法

- 此时 $a = bk + c$，有 $c = a \bmod b$
- 设 $d \mid a, d \mid b$，有 $c = a - bk$，$\frac{c}{d} = \frac{a}{d} - \frac{b}{d}k$
- 由于 d 是 a, b 的公约数，$\frac{a}{d} - \frac{b}{d}k$ 为整数，所以 $\frac{c}{d}$ 也是整数，d 也是 $b, a \bmod b$ 的公约数
- 反过来，也可以证明 $b, a \bmod b$ 的公约数也是 a, b 的公约数
- 既然两式公约数都是相同的，那么最大公约数也会相同
- 所以得到式子 $\gcd(a, b) = \gcd(b, a \bmod b)$
- 这里两个数的大小是不会增大的，那么我们也就得到了关于两个数的最大公约数的一个递归求法

欧几里得算法

```
1 int gcd(int a, int b) {
2     if (b == 0) return a;
3     return gcd(b, a % b);
4 }
```

- 递归至 b == 0（即上一步的 a % b == 0）的情况再返回值即可

欧几里得算法

- P1029 [NOIP2001 普及组] 最大公约数和最小公倍数问题
- 给定正整数 x, y，求满足 $\gcd(P, Q) = x, \operatorname{lcm}(P, Q) = y$ 的正整数 P, Q 个数
- $2 \leq x, y \leq 10^5$

欧几里得算法

- P1029 [NOIP2001 普及组] 最大公约数和最小公倍数问题
- 给定正整数 x, y，求满足 $\gcd(P, Q) = x, \operatorname{lcm}(P, Q) = y$ 的正整数 P, Q 个数
- $2 \le x, y \le 10^5$
- $\operatorname{lcm}(x, y) = \frac{xy}{\gcd(x,y)}$，$xy = \gcd(x, y) \times \operatorname{lcm}(x, y) = PQ$
- 直接枚举 P, Q 是 $O(x^2y^2)$ 的，显然会超时
- 本质上是对 xy 做因数分解，从 1 到 $\sqrt{xy}$ 枚举 P
- 如果 $Q = \frac{xy}{P}$ 是整数，且 $\gcd(P, Q) = x$，此时的 P, Q 便是一种答案
- 如果 $P \neq Q$，Q, P 也是一种答案，要特判 $P = Q$ 的情况
- 时间复杂度 $O(\sqrt{xy})$

欧几里得算法

```
1  ll gcd(ll x,ll y)
2  {
3      if(y==0) return x;
4      return gcd(y,x%y);
5  }
6  int main()
7  {
8      cin>>x>>y;
9      int ans=0;
10     for(int i=1;i<=sqrt(x*y);i++)
11     {
12         if(x*y%i==0&&gcd(i,x*y/i)==x) ans++;
13     }
14     ans*=2;
15     if(x==y) ans--;
16     cout<<ans;
17     return 0;
18 }
```

排列组合

- 排列组合是组合数学中的基础
- 排列就是指从给定个数的元素中取出指定个数的元素进行排序
- 组合则是指从给定个数的元素中仅仅取出指定个数的元素，不考虑排序
- 排列组合的中心问题是研究给定要求的排列和组合可能出现的情况总数

加法 & 乘法原理

- 加法原理：完成一个工程可以有 n 类办法，$a_i(1 \le i \le n)$ 代表第 i 类方法的数目，那么共有 $S = a_1 + a_2 + \cdots + a_n$ 种不同的完成方法
- 乘法原理：完成一个工程需要分 n 个步骤，$a_i(1 \le i \le n)$ 代表第 i 个步骤的方法数目，那么共有 $S = a_1 \times a_2 \times \cdots \times a_n$ 种不同的完成方法
- 对于乘法原理的每一个步骤，可以再结合上加法原理
- 例如选一个 $A + B$ 套餐，A 类可以选 x 种中餐和 y 种西餐之一，B 类可以选 z 种小吃之一，总选择数为 $(x + y) \times z$

排列数

- 从 n 个不同元素中，任取 $m(m \leq n)$ 个元素按照一定的顺序排成一列，叫做从 n 个不同元素中取出 m 个元素的一个排列
- 从 n 个不同元素中取出 $m(m \leq n)$ 个元素的所有排列的个数，叫做从 n 个不同元素中取出 m 个元素的排列数，用符号 A_n^m 或 P_n^m 来表示

排列数

- 排列的计算公式如下：

$$\mathrm{A}_n^m = n(n-1)(n-2)\cdots(n-m+1) = \frac{n!}{(n-m)!}$$

  $n!$ 代表 n 的阶乘，即 $6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6$

- 公式可以这样理解：n 个人选 m 个人来排队，第一个位置可以选 n 个，第二位置可以选 $n-1$ 个，以此类推，第 m 个（最后一个）可以选 $n-m+1$ 个

排列数

- 全排列：n 个人全部来排队，队长为 n。第一个位置可以选 n 个，第二位置可以选 $n-1$ 个，以此类推得：

$$\mathrm{A}_n^n = n(n-1)(n-2)\cdots 3\times 2\times 1 = n!$$

- 全排列是排列数的一个特殊情况

组合数

- 从 n 个不同元素中，任取 $m(m \leq n)$ 个元素组成一个集合，叫做从 n 个不同元素中取出 m 个元素的一个组合。相比于排列，组合是无序的
- 从 n 个不同元素中取出 $m(m \leq n)$ 个元素的所有组合的个数，叫做从 n 个不同元素中取出 m 个元素的组合数，用符号 $\binom{n}{m}$ 或 C_n^m 来表示，读作「n 选 m」

组合数

- 组合数计算公式：

$$\binom{n}{m} = \frac{\mathrm{A}_n^m}{m!} = \frac{n!}{m!(n-m)!}$$

- 如何理解上述公式？考虑 n 个人选 $m(m \leq n)$ 个出来，不排队、不在乎顺序
- 如果在乎顺序那么就是 A_n^m，如果不在乎那么就要除掉重复，同样选出来的 m 个人，他们还要「全排」得 $m!$，所以要再除以 $m!$
- 组合数也被称为「二项式系数」
- 特别地，规定当 $m > n$ 时，$\mathrm{A}_n^m = \binom{n}{m} = 0$

插板法

- 插板法是用于求一类给相同元素分组的方案数的一种技巧，也可以用于求一类线性不定方程的解的组数
- 问题一：现有 n 个完全相同的元素，要求将其分为 k 组，保证每组至少有一个元素，一共有多少种分法？

插板法

- 插板法是用于求一类给相同元素分组的方案数的一种技巧，也可以用于求一类线性不定方程的解的组数
- 问题一：现有 n 个完全相同的元素，要求将其分为 k 组，保证每组至少有一个元素，一共有多少种分法？
- 考虑拿 $k-1$ 块板子插入到 n 个元素两两形成的 $n-1$ 个空里面
- 因为元素是完全相同的，所以答案就是 $\binom{n-1}{k-1}$
- 本质是求 $x_1+x_2+\cdots+x_k=n$ 的正整数解的组数

插板法

- 问题二：如果问题变化一下，每组允许为空呢？

插板法

- 问题二：如果问题变化一下，每组允许为空呢？
- 显然此时没法直接插板了，因为有可能出现很多块板子插到一个空里面的情况，非常不好计算
- 我们考虑创造条件转化成有限制的问题一，先借 k 个元素过来，在这 $n+k$ 个元素形成的 $n+k-1$ 个空里面插板，答案为

$$\binom{n+k-1}{k-1}=\binom{n+k-1}{n}$$

- 开头我们借来了 k 个元素，用于保证每组至少有一个元素，插完板之后再把这 k 个借来的元素从 k 组里面拿走
- 本质是求 $x_1+x_2+\cdots+x_k=n$ 的非负整数解的组数（即要求 $x_i\geq 0$）

插板法

- 例题：有 10 个完全相同的球，要放进 3 个不同的盒子里，第 i 个盒子里需要至少有 i 个球，求总方案数

插板法

- 例题：有 10 个完全相同的球，要放进 3 个不同的盒子里，第 i 个盒子里需要至少有 i 个球，求总方案数
- 相比允许为空时的借球，现在应该先把球拿出去
- 问题等价于，有 7 个完全相同的球，要放进 3 个盒子里，不允许盒子为空，求总方案数
- 由插板法得答案为

$$\binom{7-1}{3-1} = \binom{6}{2} = 15$$

组合数的性质

- 组合数的对称性：

$$\binom{n}{m} = \binom{n}{n-m}$$

- 由定义导出的递推式：

$$\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$$

- 组合数的递推式（杨辉三角）：

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1}$$

组合数的性质

- 组合数的拆分：

$$\sum_{i=0}^{k} \binom{n}{i} \binom{m}{k-i} = \binom{n+m}{k}$$

- 考虑其组合意义，也就是从 m 个男生 n 个女生中选出 k 个人的方案数
- 上式也叫范德蒙德（Vandermode）卷积

预处理组合数

- 由递推式可以得出组合数的 $O(n^2)$ 求法

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1}$$

```
for(int i=0;i<=n;i++)
    c[i][0]=1;
for(int i=1;i<=n;i++)
    for(int j=1;j<=i;j++)
        c[i][j]=(c[i-1][j]+c[i-1][j-1])%p;
```

二项式定理

- 二项式定理阐明了一个展开式的系数：

$$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^{n-i} b^i$$

- $\binom{n}{i}$ 是 n 次二项式展开后第 i 项的二项式系数，也对应杨辉三角第 n 行第 i 个
- $n=2$ 时便是经典的 $(a+b)^2 = a^2 + 2ab + b^2$
- 证明采用归纳法，这里不作详细证明

二项式定理

- 二项式定理可以推出组合数更多的性质：

$$(a-b)^n=\sum_{i=0}^{n}(-1)^i\binom{n}{i}a^{n-i}b^i \tag{1}$$

$$2^n=(1+1)^n=\sum_{i=0}^{n}\binom{n}{i} \tag{2}$$

$$0=(1-1)^n=\sum_{i=0}^{n}(-1)^i\binom{n}{i} \tag{3}$$

二项式定理

- P1313 [NOIP2011 提高组] 计算系数
- 给定多项式 $(ax+by)^k$，求多项式展开后 $x^n \times y^m$ 项的系数，答案对 10007 取模
- $0 \leq n, m < k \leq 10^3$，$n+m=k$，$0 \leq a, b \leq 10^6$

二项式定理

- P1313 [NOIP2011 提高组] 计算系数
- 给定多项式 $(ax+by)^k$，求多项式展开后 $x^n \times y^m$ 项的系数，答案对 10007 取模
- $0 \le n, m < k \le 10^3$，$n+m=k$，$0 \le a, b \le 10^6$
- 根据二项式定理：

$$(ax+by)^k = \sum_{i=0}^{k} \binom{k}{i} (ax)^i (by)^{k-i} = \sum_{i=0}^{k} \binom{n}{i} a^i b^{k-i} x^i y^{k-i}$$

- 那么 $x^n \times y^m$ 项的系数为 $\binom{k}{n} a^n b^m$

二项式定理

```
ll pw(ll x,ll y){
    ll ans=1;
    for(ll i=1;i<=y;i++)
        ans=ans*x%10007;
    return ans;
}
int main(){
    cin>>a>>b>>k>>n>>m;
    for(ll i=0;i<=k;i++)
        f[i][0]=1;
    for(ll i=2;i<=k+1;i++)
        for(ll j=1;j<=i;j++)
            f[i][j]=(f[i-1][j-1]+f[i-1][j])%10007;
    ll ans=(pw(a,n)*pw(b,m))%10007*f[k][n]%10007;
    cout<<ans;
    return 0;
}
```

总结

- 本节课以讲授数学工具为主，仅用到对应知识的纯练习题并不多
- 如有兴趣，可以自行学习如下知识点：
  - 高精度 gcd（更相减损术）
  - 扩展欧几里得算法（exgcd）
  - 乘法逆元
  - Lucas 定理