数学专题选讲(积性函数与组合计数)

2025 CEIT 算法集训队暑期集训 #4

Orange

沈阳师范大学，人工智能学院，CEIT LAB

2025/08/19

目录

积性函数的定义

定义

对于函数f，若函数满足对于任意互质的正整数x和y满足$f(xy)=f(x)f(y)$，则称f为**积性函数**。

定义

特别的，若x和y不互质，也满足$f(xy)=f(x)f(y)$，则称f为**完全积性函数**。

理解

- 积性函数：$\gcd(x,y)=1 \Rightarrow f(xy)=f(x)f(y)$
- 完全积性函数：对任意x,y都有$f(xy)=f(x)f(y)$

常见的积性函数

- $\mathbf{1}(n) = 1$ （常数函数，完全积性）
- $\mathrm{id}(n) = n$ （恒等函数，完全积性）
- $\varepsilon(n) = [n = 1]$ （单位函数）
- $\varphi(n) = \sum_{i=1}^{n}[\gcd(i, n) = 1]$ （欧拉函数）
- $d(n) = \sum_{i=1}^{n}[i|n]$ （因数个数函数）
- $\sigma(n) = \sum_{d|n} d$ （因数和函数）
- $\mu(n)$ （莫比乌斯函数）

重要性质

若$f(x)$和$g(x)$都是积性函数，则$h(x) = f(x)g(x)$也是积性函数。

积性函数的求解方法一：质因数分解

设f为某个积性函数，假设$x = p_1^{c_1} p_2^{c_2} p_3^{c_3} \dots p_k^{c_k}$，则：

$$f(x) = f(p_1^{c_1}) f(p_2^{c_2}) f(p_3^{c_3}) \dots f(p_k^{c_k})$$

算法流程

1. 对x进行质因数分解
2. 分别计算$f(p_i^{c_i})$
3. 将结果累乘得到$f(x)$

时间复杂度： $O(\sqrt{n} \times O(f))$

适用场景

单次查询，或者查询次数较少的情况。

积性函数的求解方法二：欧拉筛

在欧拉筛的过程中，我们会遇到以下情况：

- **若i是质数**：此时$f(i)$在大多数情况是好求的，直接计算即可
- **若筛非质数**：考虑i被最小质因子p筛去的两种情况：
  - 若p在i中第一次出现，则$f(i \times p) = f(i)f(p)$
  - 若p不是第一次出现，则$f(i \times p) = \frac{f(i)}{f(p^c)} \times f(p^{c+1})$，其中$c$为$i$中$p$的次数

时间复杂度：$O(n)$

适用场景

需要预处理大量数的积性函数值的情况。

例题1：因数个数查询

题目描述

给定q次询问，每次询问一个正整数x，请回答x的所有因数个数。

数据范围： $q \le 10^5, x \le 10^7$

时间限制： 1秒

例题1：解题思路

问题分析

- 朴素做法：对每个x都进行因数分解，时间复杂度$O(q\sqrt{x})$
- 当$q = 10^5, x = 10^7$时，总复杂度约为$10^5 \times \sqrt{10^7} \approx 3 \times 10^8$
- 显然会超时，需要优化

关键观察

因数个数函数$d(x)$是积性函数！

对于$x = p_1^{c_1} p_2^{c_2} \cdots p_k^{c_k}$，有：

$$d(x) = d(p_1^{c_1}) \cdot d(p_2^{c_2}) \cdots d(p_k^{c_k}) = \prod_{i=1}^{k} (c_i + 1)$$

这是因为$p_i^{c_i}$的因数为$\{1, p_i, p_i^2, \ldots, p_i^{c_i}\}$，共$c_i + 1$个。

例题1：详细解法

欧拉筛求解积性函数

利用积性函数的性质，可以在欧拉筛的过程中同时计算$d(x)$：

- **质数情况：** 若i是质数，则$d(i)=2$（因数为1和i）
- **合数情况：** 设i被最小质因子p筛去
  - 若$p\nmid i$：则$\gcd(i,p)=1$，$d(i\cdot p)=d(i)\cdot d(p)$
  - 若$p\mid i$：设$i=p^c\cdot m$，其中$\gcd(p,m)=1$
    则$d(i\cdot p)=d(p^{c+1})\cdot d(m)=(c+2)\cdot d(m)$
    而$d(i)=(c+1)\cdot d(m)$，所以$d(i\cdot p)=\frac{c+2}{c+1}\cdot d(i)$

算法复杂度

- 预处理：$O(n)$，其中$n=10^7$
- 单次查询：$O(1)$
- 总复杂度：$O(n+q)=O(10^7)$，完全可以接受

例题1：代码实现

```
const int MAXN = 1e7 + 5;
int d[MAXN], cnt[MAXN]; // d[i]表示i的因数个数，cnt[i]表示i的最小质因子的次数
bool is_prime[MAXN];
vector<int> primes;

void sieve() {
    fill(is_prime, is_prime + MAXN, true);
    d[1] = 1;
    for (int i = 2; i < MAXN; i++) {
        if (is_prime[i]) {
            primes.push_back(i);
            d[i] = 2; // 质数的因数个数为2
            cnt[i] = 1;
        }
        for (int j = 0; j < primes.size() && i * primes[j] < MAXN; j++) {
            int p = primes[j];
            is_prime[i * p] = false;
            if (i % p == 0) {
                cnt[i * p] = cnt[i] + 1;
                d[i * p] = d[i] / (cnt[i] + 1) * (cnt[i * p] + 1);
                break;
            } else {
                cnt[i * p] = 1;
                d[i * p] = d[i] * d[p];
            }
        }
    }
}
```

例题2：华华给月月出题

题目描述

华华给月月出了一道数学题：计算 $\bigoplus_{i=1}^{N} i^N \bmod (10^9+7)$

数据范围： $1 \leq N \leq 1.3 \times 10^7$

时间限制： 1秒

例题2：解题思路

问题分析

- 朴素做法：对每个i计算i^N，时间复杂度$O(N\log N)$
- 当$N = 1.3 \times 10^7$时，复杂度约为3×10^8，可能超时
- 需要利用数学性质进行优化

关键观察

定义函数$f(x) = x^N$，这是一个完全积性函数!

- 对于任意正整数a, b，都有$f(ab) = (ab)^N = a^N b^N = f(a)f(b)$
- 这意味着我们可以用欧拉筛的思想来计算所有$f(i)$的值
- 关键是避免重复的幂运算

例题2：详细解法

欧拉筛优化

核心思想：利用完全积性函数的性质减少幂运算次数

- **质数情况**：若i是质数，则$f(i) = i^N$，需要用快速幂计算
- **合数情况**：若i被最小质因子p筛去，则$f(i \cdot p) = f(i) \cdot f(p)$

复杂度分析

关键引理：质数定理告诉我们，小于等于N的质数个数约为$\frac{N}{\ln N}$

- 质数的幂运算：$O\left(\frac{N}{\ln N} \log N\right) = O(N)$
- 合数的乘法运算：$O(N)$
- 总复杂度：$O(N)$，相比朴素算法有显著提升

实现要点

- 使用欧拉筛模板，在筛质数的同时计算f值
- 注意取模运算，防止溢出

例题3：Bash Plays with Functions (Codeforces 757E)

题目描述

定义函数$f_0(n)$为满足$pq = n$且$\gcd(p, q) = 1$的有序对(p, q)的个数。

定义递推关系：

$$f_{r+1}(n) = \sum_{pq=n} \frac{f_r(p) + f_r(q)}{2}$$

给定q次询问，每次询问$f_r(n) \bmod (10^9 + 7)$的值。

数据范围：$1 \le n, r, q \le 10^6$

例题3：思路分析

理解$f_0(n)$

对于$n = p_1^{c_1} p_2^{c_2} \cdots p_k^{c_k}$，要使$\gcd(p, q) = 1$且$pq = n$：

- 每个质因子$p_i^{c_i}$只能完全分给p或q中的一个
- 有k个不同的质因子，每个有2种选择
- 因此$f_0(n) = 2^{\omega(n)}$，其中$\omega(n)$是n的不同质因子个数

关键观察

$f_0(n)$是积性函数！

证明：设$\gcd(a, b) = 1$，则$\omega(ab) = \omega(a) + \omega(b)$

所以$f_0(ab) = 2^{\omega(ab)} = 2^{\omega(a)+\omega(b)} = 2^{\omega(a)} \cdot 2^{\omega(b)} = f_0(a) \cdot f_0(b)$

例题3：递推关系分析

递推式变形

原递推式：$f_{r+1}(n) = \sum_{pq=n} \frac{f_r(p)+f_r(q)}{2}$
关键变形：$f_{r+1}(n) = \sum_{d|n} f_r(d)$
证明思路：

- 对于每个因子$d|n$，令$p = d, q = \frac{n}{d}$
- 在原式中，$(d, \frac{n}{d})$和$(\frac{n}{d}, d)$各贡献$\frac{f_r(d)+f_r(\frac{n}{d})}{2}$
- 总贡献为$f_r(d) + f_r(\frac{n}{d})$，对所有$d|n$求和即得新递推式

定理 (重要性质)

若f是积性函数，则$g(n) = \sum_{d|n} f(d)$也是积性函数。

推论：所有的$f_r(n)$都是积性函数！

例题3：详细解法

利用积性函数性质

对于质数幂p^c，有递推关系：

$$f_r(p^c) = \sum_{i=0}^{c} f_{r-1}(p^i)$$

边界条件： $f_0(p^c) = 2$（当$c \geq 1$时），$f_0(1) = 1$

预处理策略

- 由于$n \leq 10^6$，质数幂的指数$c \leq \lfloor \log_2(10^6) \rfloor = 19$
- 预处理所有质数$p \leq 10^6$和所有$r \leq 10^6$的$f_r(p^c)$值
- 对于查询$f_r(n)$，将n分解为质数幂的乘积，利用积性函数性质计算

复杂度分析

加法原理和乘法原理

定理 (加法原理)

做某件事情有几种选择，每种选择的方案数之和就是做这件事情的方案数。

定理 (乘法原理)

做某件事情分为几步，每步的方案数是独立的，则它们的积就是做这件事情的方案数。

例子

加法原理：从A地到B地，可以坐火车（3种方案）或飞机（2种方案），总共$3+2=5$种方案。
乘法原理：从A地经C地到B地，A到C有3种方案，C到B有2种方案，总共$3\times 2=6$种方案。

排列

定义 (排列)

设集合S是一个n元集，序列$a_1, a_2, \ldots, a_r$满足$a_i \in S$，且序列中的元素两两不同，求有多少个这样的序列。

符号表示：

- 下降幂：$n^{\underline{r}} = n \cdot (n-1) \cdot (n-2) \cdots (n-r+1)$
- 上升幂：$n^{\overline{r}} = n \cdot (n+1) \cdot (n+2) \cdots (n+r-1)$
- 阶乘：$n! = n \cdot (n-1) \cdots 1$，$0! = 1$
- 排列数：$P(n,r) = A_n^r = \frac{n!}{(n-r)!}$

注意

$n^{\underline{r}} = (n-r+1)^{\overline{r}}$

组合

定义 (组合)

设集合S是一个n元集，集合$\{a_1, a_2, \ldots, a_r\}$满足$a_i \in S$，求有多少个这样的集合。

符号表示：

$$\binom{n}{r} = C_n^r = \frac{n^{\underline{r}}}{r!} = \frac{n!}{(n-r)!r!}$$

常用恒等式：

- $\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m}$ （帕斯卡恒等式）
- $\binom{n}{m} = \frac{n}{m}\binom{n-1}{m-1}$
- $\binom{n}{m} = \frac{n-m+1}{m}\binom{n}{m-1}$
- $\binom{n}{k} = \binom{n}{n-k}$ （对称性）

组合数的重要性质

定理 (二项式定理)

$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$

重要推论:

- $\sum_{k=0}^{n} \binom{n}{k} = 2^n$ （令$x = y = 1$）
- $\sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0$ （令$x = 1, y = -1$，$n > 0$）
- $\sum_{k=0}^{n} k \binom{n}{k} = n \cdot 2^{n-1}$
- $\sum_{k=0}^{n} \binom{n}{k}^2 = \binom{2n}{n}$ （范德蒙德恒等式特例）

定理 (范德蒙德恒等式)

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

隔板法

问题：求不定方程$x_1+x_2+\cdots+x_k=n$的解的数量，其中x_i为正整数。

隔板法思想

- 将n个相同的球排成一行
- 在$n-1$个空隙中选择$k-1$个位置放隔板
- 隔板将球分成k组，每组至少有1个球
- 答案为$\binom{n-1}{k-1}$

变形问题：

- 若$x_i \geq a_i$，则令$y_i = x_i - a_i$，转化为$\sum y_i = n - \sum a_i + k$，$y_i \geq 0$。此时若令$\sum a_i = 0$，则可得到方程非负解的个数为$\binom{n+k-1}{k-1}$。
- 若$x_1+x_2+\cdots+x_k \leq n$，则引入松弛变量$x_{k+1}$使得$x_1+x_2+\cdots+x_k+x_{k+1}=n$，此时$x_{k+1}\geq 0$，套用之前的结论求解即可。

网络路径计数

问题： 在$n \times m$的网格图上，从$(0,0)$走到(n,m)，每次只能向右走或向上走，求方案数。

分析

- 总共需要走$n+m$步
- 其中向右走n步，向上走m步
- 问题转化为：在$n+m$个位置中选择n个位置向右走
- 答案为$\binom{n+m}{n} = \binom{n+m}{m}$

扩展： 如果有障碍物怎么办？

- 使用容斥原理
- 或者动态规划

例题1：Lucky Numbers (Codeforces 630C)

题目描述

Petya喜欢幸运数字。我们称一个数字为幸运数字，当且仅当它的所有数位都是7或8。
现在Petya想知道：有多少个位数不超过n的正整数是幸运数字？
数据范围： $1 \leq n \leq 55$

样例

输入：

2

输出：

6

解释： 1位数有7, 8（2个）；2位数有77, 78, 87, 88（4个）；总共6个。

例题1：解题思路

问题分析

- 幸运数字的每一位都只能是7或8
- 需要统计位数从1到n的所有幸运数字个数
- 这是一个典型的乘法原理应用

按位数分类计算

- **1位数：** 只有7和8，共$2^1 = 2$个
- **2位数：** 每位都可以选7或8，共$2^2 = 4$个
- **3位数：** 每位都可以选7或8，共$2^3 = 8$个
- **一般地，k位数：** 共2^k个

最终答案

总答案为：$\sum_{i=1}^{n} 2^i = 2^1 + 2^2 + \cdots + 2^n = 2(2^n - 1) = 2^{n+1} - 2$

公式推导： 利用等比数列求和公式$\sum_{i=1}^{n} 2^i = 2 \cdot \frac{2^n-1}{2-1} = 2^{n+1} - 2$

例题2：Far Relative's Birthday Cake (Codeforces 629A)

题目描述

Polycarp的远房亲戚过生日，他准备了一个$n \times n$的方形蛋糕。蛋糕被切成n^2个小方块，每个方块要么有樱桃（用'C'表示），要么没有（用'.'表示）。

如果两个樱桃在同一行或同一列，它们就会产生"幸福"。求总共有多少对樱桃会产生幸福？

数据范围： $1 \leq n \leq 100$

例题2：解题思路

分类计算

设第i行有r_i个樱桃，第j列有c_j个樱桃：

- **同行的樱桃对：** 第i行内部可以选择$\binom{r_i}{2}$对樱桃
- **同列的樱桃对：** 第j列内部可以选择$\binom{c_j}{2}$对樱桃
- **总答案：** $\sum_{i=1}^{n}\binom{r_i}{2}+\sum_{j=1}^{n}\binom{c_j}{2}$

实现细节

- 遍历整个网格，统计每行每列的樱桃数量
- 对于每行每列，如果樱桃数≥ 2，则贡献$\binom{k}{2}=\frac{k(k-1)}{2}$
- 时间复杂度：$O(n^2)$

例题3：Beautiful Numbers (Codeforces 300C)

题目描述

给定两个不同的数字a, b（$1 \le a < b \le 9$）。
好数： 只包含数字a和b的正整数。
极好数： 既是好数，且其数位和也是好数的正整数。
求有多少个恰好n位的极好数？答案对$10^9 + 7$取模。
数据范围： $1 \le a < b \le 9$，$1 \le n \le 10^6$

例题3：解题思路

问题分析

- 需要构造n位数，每位只能是a或b
- 构造的数的数位和也必须只包含数字a和b
- 这是组合数学与数字性质的结合

关键观察

设n位数中有i个数字a，$(n-i)$个数字b：

- **数位和**：$S = i \cdot a + (n-i) \cdot b = n \cdot b + i \cdot (a-b)$
- **排列方案数**：在n个位置中选择i个位置放数字a，方案数为$\binom{n}{i}$
- **合法条件**：S必须是好数（只包含数字a和b）

例题3：详细解法

算法流程

1. 枚举i从0到n（a的个数）
2. 计算对应的数位和$S = i \cdot a + (n - i) \cdot b$
3. 判断S是否为好数（只包含数字a和b）
4. 如果S是好数，则累加$\binom{n}{i}$到答案中

判断好数的方法

检查数字S的每一位是否都是a或b:

- 不断取S的个位数字，检查是否为a或b
- 如果某一位不是a或b，则S不是好数
- 时间复杂度：$O(\log S)$

例题3：Makes And The Product (Codeforces 817B)

题目描述

给定序列a_i，求有多少个三元组(i, j, k)满足$i < j < k$且$a_i a_j a_k$是所有三个元素的乘积之中最小的。

数据范围：$3 \le n \le 10^5, 1 \le a_i \le 10^9$

例题3：解题思路

问题分析

- 首先需要找到所有三个元素乘积中的最小值
- 由于都是正整数，对序列排序后前三项乘积即为最小值
- 设最小乘积为$p = a_1 a_2 a_3$（排序后）
- 问题转化为：有多少个(i, j, k)满足$i < j < k$且$a_i a_j a_k = p$

例题3：分类讨论

计数方法

设$cnt[x]$表示序列中值为x的元素个数，分以下情况：

- **情况1：** $cnt[a_1] \geq 3 \rightarrow$ 答案为$\binom{cnt[a_1]}{3}$
- **情况2：** $cnt[a_1] = 2$或$(cnt[a_1] = 1 \land cnt[a_2] = 1) \rightarrow$ 答案为$cnt[a_3]$
- **情况3：** $cnt[a_1] = 1 \land cnt[a_2] > 1 \rightarrow$ 答案为$\binom{cnt[a_2]}{2}$

例题4：The World is a Theatre (Codeforces 131C)

题目描述

从n个男孩和m个女孩中挑选一个t人小组，要求组内至少有1个女孩和4个男孩，求方案数。

数据范围： $n, m \leq 30, t \leq n + m$

例题4：解题思路

问题分析

- 需要选择t个人，其中至少1个女孩，至少4个男孩
- 可以枚举女孩的个数k，则男孩个数为$t-k$
- 约束条件：$k \geq 1$且$t-k \geq 4$，即$1 \leq k \leq t-4$

组合数求解

对于固定的女孩个数k：

- 从m个女孩中选k个：$\binom{m}{k}$
- 从n个男孩中选$t-k$个：$\binom{n}{t-k}$
- 该情况的方案数：$\binom{m}{k} \cdot \binom{n}{t-k}$

最终答案：

$$\sum_{k=1}^{t-4} \binom{m}{k} \binom{n}{t-k}$$

例题5：The Intriguing Obsession (Codeforces 869C)

题目描述

有3个群岛A、B、C，分别有a、b、c个小岛。在其中建桥，要求一个群岛内的任意两个小岛至少需要跨过3座桥才能到达。求建桥方案数，答案对998244353取模。

数据范围： $a, b, c \leq 5000$

例题5：解题思路

约束分析

- 群岛内任意两岛至少跨3座桥→ 群岛内不能有桥
- 如果A群岛的岛u与B群岛的岛v有桥，则u和v都不能再连其他桥
- 不同群岛对之间的桥互不影响

关键观察

- 问题可以分解为三个独立的子问题
- 分别计算AB、AC、BC群岛间的建桥方案数
- 最终答案为三者的乘积

例题5：详细解法

两群岛间建桥方案

考虑A、B两群岛间建k座桥：

- 在A中选k个端点：$\binom{a}{k}$
- 在B中选k个端点：$\binom{b}{k}$
- k个端点的配对方案：$k!$

该情况方案数：$\binom{a}{k}\binom{b}{k}k!$

最终公式

A、B间总方案数：

$$f_{ab} = \sum_{k=0}^{\min(a,b)} \binom{a}{k}\binom{b}{k}k!$$

例题5：详细解法

最终答案：

$$f_{ab} \cdot f_{ac} \cdot f_{bc}$$

例题6：Count the Arrays (Codeforces 1312D)

题目描述

求满足如下条件的数列个数，答案对998244353取模：

- 长度为n
- 每个元素值域为1到m
- 有且仅有一对相等的元素
- 存在下标i，满足$a_1 < \cdots < a_{i-1} < a_i > a_{i+1} > \cdots > a_n$

数据范围： $n, m \leq 2 \times 10^5$

例题6：解题思路

结构分析

- 序列形如：$a_1 < \cdots < a_{i-1} < a_i > a_{i+1} > \cdots > a_n$
- a_i必须是序列中的最大值
- 有且仅有一对相等元素$\rightarrow$ 序列中有$n-1$个不同的数

关键观察

- 峰值位置的数必须是最大值
- 重复的数不能是最大值
- 重复的数必须分布在峰值两侧

例题6：详细解法

分步计算

步骤1： 选择$n-1$个不同的数$\to \binom{m}{n-1}$
步骤2： 选择重复出现的数（不能是最大值）$\to n-2$种选择
步骤3： 安排数的位置

- 最大值固定在峰值位置
- 重复的数一个在左侧，一个在右侧
- 其余$n-3$个数可选择左侧或右侧$\to 2^{n-3}$

最终答案

$$\binom{m}{n-1} \cdot (n-2) \cdot 2^{n-3}$$

Thank You!

参考文献

1 积性函数手稿 - CEIT算法集训队内部资料
2 组合数学课程 - emofunc, 2021
3 【组合数学】排列组合（一）- 三金家的Y同学
4 【组合数学】排列组合（二）- 三金家的Y同学
5 Codeforces题目集 - 各类组合计数经典题目