浅谈容斥原理

成都七中 王迪

摘 要

本文从计数问题中的容斥原理出发，得出容斥原理的形式，通过一些例题探究了使用它解题的思维方式，然后研究了容斥原理的推广，对容斥原理的一般化作出尝试，最后总结了容斥原理及其运用过程中所体现的思想、方法。

1 引言

在一类组合计数的问题中，我们需要对一些集合的并或交中元素的个数进行统计，而对于这种问题，容斥原理是一种通用的解法，所以在本文的前半部分，我们将探究容斥原理的形式，用容斥原理解决计数问题的分析方法，以及若干有趣的例题。

而容斥原理不仅可以解决组合计数问题，在本文的后半部分，我们将对容斥原理进行推广，可以解决一些数论和概率论中的问题，而通过分析不同问题中容斥原理的形式，我们可以将容斥原理一般化，从更高的层面理解容斥原理。

2 容斥原理

在这一小节中，我们会从一些组合计数问题出发，得出容斥原理的形式并给出证明，然后通过一些例题得出用容斥原理解题的思维方法。

2.1 预备知识

考虑一个简单的问题：某班有 a 个人擅长唱歌，b 个人擅长画画，c 个人既擅长唱歌也擅长画画，问多少人有至少一种特长？

通过画文氏图的方法，很容易得出此问题的答案：$a+b-c$。

我们可以得出该问题的一般形式：设一个有限集为 U，U 中元素有两种性质 P_1 和 P_2，而满足 P_1 性质的元素组成集合 S_1，满足 P_2 性质的元素组成集合 S_2，那么上面的问题相当于是求至少满足两种性质之一的元素个数，可以表示成这样：

$$|S_1 \cup S_2| = |S_1| + |S_2| - |S_1 \cap S_2|$$

其中 $|S|$ 表示集合 S 中的元素个数。

我们考虑 U 中元素有三种性质 P_1, P_2, P_3，对应的子集是 S_1, S_2, S_3，仍然可以通过画文氏图的方法，得到下面的等式：

$$|S_1 \cup S_2 \cup S_3| = |S_1| + |S_2| + |S_3| - |S_1 \cap S_2| - |S_1 \cap S_3| - |S_2 \cap S_3| + |S_1 \cap S_2 \cap S_3|$$

注意到，我们把**求并集中元素个数**转化成了**求交集中元素个数**，这一步体现了**转化**的思想。

一般地，设 U 中元素有 n 种不同的性质，第 i 种性质称为 P_i，满足 P_i 的元素组成集合 S_i，那么

定理 2.1.1. *满足 $P_1, P_2, \cdots, P_n$ 中至少一个性质的 U 中元素的个数是*

$$\begin{aligned}|S_1 \cup S_2 \cup \cdots \cup S_n| &= \sum_i |S_i| - \sum_{i<j} |S_i \cap S_j| + \sum_{i<j<k} |S_i \cap S_j \cap S_k| - \cdots \\ &\quad \cdots + (-1)^{n-1} |S_1 \cap S_2 \cap \cdots \cap S_n|\end{aligned}$$

证明. 考虑一个处于 $\bigcup_{i=1}^{n} S_i$ 中的元素 x，它所属 m 个集合 $T_1, T_2, \cdots, T_m$，那么我们计算一下上式右边统计的 x 个数 cnt：

$$\begin{aligned}cnt &= |\{T_i\}| - |\{T_i \cap T_j | i < j\}| + \cdots + (-1)^{m-1} |\{T_1 \cap T_2 \cap \cdots \cap T_m\}| \\ &= C_m^1 - C_m^2 + \cdots + (-1)^{m-1} C_m^m \\ &= -\left(\left(\sum_{i=0}^{m} (-1)^i C_m^i\right) - C_m^0\right) \\ &= -\left((1-1)^m - 1\right) \\ &= 1\end{aligned}$$

结合二项式定理即可证明容斥原理的正确性。 □

至此，我们已经知道了用容斥原理计算集合的并中元素的数目的方法。稍加变形我们就能用容斥原理计算集合的交中元素的数目。

我们用 $\bigcap_{i=1}^{n} S_i$ 表示需要计数的交集，令 $\overline{S}$ 表示集合 S 关于全集 U 的补集，那么 $\overline{S_i}$ 就表示不满足性质 P_i 的集合。考虑到需要计数的是满足 $P_1, P_2, \cdots, P_n$ 的元素，我们进行一步**补集转化**，求不满足至少一个性质的 U 中的元素个数，即 $\bigcup_{i=1}^{n} \overline{S_i}$，这个集合的计数方式就和之前类似，所以

定理 2.1.2. *满足 $P_1, P_2, \cdots, P_n$ 中所有性质的 U 中元素的个数是*

$$|S_1 \cap S_2 \cap \cdots \cap S_n| = |U| - |\overline{S_1} \cup \overline{S_2} \cup \cdots \cup \overline{S_n}|$$

综上所述，我们可以利用容斥原理在**求并集元素个数**和**求交集元素个数**这两个问题间**互相转化**，这提示我们，用容斥原理解题，是一个**转换角度**的思维方式。

2.2 经典问题

我们通过几个组合计数的经典题目，来探究如何应用容斥原理。

2.2.1 不定方程非负整数解计数

问题 2.2.1. 考虑不定方程

$$x_1 + x_2 + \cdots + x_n = m$$

和 n 个限制条件 $x_i \leq b_i$，其中 m 和 b_i 都是非负整数，求该方程的非负整数解的数目。

在解决这个问题之前，这里不加证明地给出一个结论：

定理 2.2.1. *不定方程 $x_1 + x_2 + \cdots + x_n = m$ 的非负整数解数目为 C_{m+n-1}^{n-1}。*

在应用容斥原理前，我们需要找出全集 U，以及刻画 U 中元素的 P_i。

- U 是满足 $x_1 + x_2 + \cdots + x_n = m$ 的所有非负整数解；
- 对于每个变量 i，都对应一个 P_i，而 P_i 代表的性质是 $x_i \leq b_i$。

设满足 P_i 的所有解组成集合 S_i，那么我们需要求解的值就是：$|\bigcap_{i=1}^{n} S_i|$。

由之前的知识我们可以写出：$|\bigcap_{i=1}^{n} S_i| = |U| - |\bigcup_{i=1}^{n} \overline{S_i}|$。而 $|U|$ 的值可以由定理 2.2.1计算，我们着重考虑后面的部分，而这正是之前容斥原理的一般形式！

通过展开 $\bigcup_{i=1}^{n} \overline{S_i}$，问题转化成：对于某几个特定的 $\overline{S_i}$，求解满足这些条件的解的数目。一般地，给出 $1 \le i_1 < i_2 < \cdots < i_t \le n$，求 $|\bigcap_{k=1}^{t} \overline{S_{i_k}}|$。

考虑 $\overline{S_{i_k}}$ 的含义，即满足 $x_{i_k} \ge b_{i_k} + 1$ 的解的数目。而对每一个 k，都要满足这个条件，即**部分变量有下界限制**，我们可以在方程的右边减去下界和 $\sum_{i=1}^{k} (b_{i_k} + 1)$，那么新方程的解与我们要求的解是一一对应的！而新方程的每个变量都没有上下界限制，所以同样可以用定理 2.2.1 求出。

于是我们只需要枚举 $\{\overline{S_1}, \overline{S_2}, \cdots, \overline{S_n}\}$ 的非空子集，进行容斥原理的计算即可。

考虑解题过程，我们先是把问题写成集合的形式，找出全集 U，以及我们的解需要满足的性质 P_i，然后写出需要求值的式子，用容斥原理进行展开，于是我们可以着眼局部，这时的限制数就大大减少，成为一个个可解的问题，最后我们把答案合并起来就可以了。

2.2.2 错位排列计数

问题 2.2.2. 称一个长度为 n 的排列 p 为错位排列，当且仅当对所有的 $1 \le i \le n$，都满足 $p_i \ne i$。给出 n，求长度为 n 的错位排列的数目。注意排列中 1 到 n 的整数都恰好出现 1 次。

同上题，我们首先分析全集 U 和性质 P_i：

- U 表示长度为 n 的所有排列；
- 对于每个位置 i，都对应一个 P_i，而 P_i 代表的性质是 $p_i \ne i$。

同样设满足 P_i 的解组成集合 S_i，那么我们需要求的值仍是 $|\bigcap_{i=1}^{n} S_i|$！

于是用同样的处理方法，我们写出 $|\bigcap_{k=1}^{t} \overline{S_{i_k}}|$，我们考虑 $\overline{S_{i_k}}$ 的含义，即 $p_i = i$ 的排列数目，而对每一个 k，都确定了排列中一个位置的数，所以共有 t 个位置的数被确定了，而其他位置是没有限制的，所以对应的答案就是 $(n-t)!$。

进一步可以推出，只要我们枚举的 $\{\overline{S_i}\}$ 的子集的大小一样，它们对答案的

贡献也是一样的！设长度为 n 的错位排列数是 D_n，那么我们有：

$$\begin{aligned}D_n &= n! - \sum_{t=1}^{n}(-1)^{t-1}\sum_{i_1<i_2<\cdots<i_t}(n-t)! \\ &= n! + \sum_{t=1}^{n}(-1)^t C_n^t (n-t)! \\ &= n! + \sum_{t=1}^{n}(-1)^t \frac{n!}{t!} \\ &= n!\sum_{t=0}^{n}\frac{(-1)^t}{t!}\end{aligned}$$

由此我们发现，用容斥原理解决问题的时间复杂度不一定是指数级，我们可以对一些对答案贡献一致的情况进行合并，这样仍能得出高效的算法。

另外，错位排列数 D_n 也有递推的方法，有兴趣的同学可以另行探究。

2.3 例题解析

下面通过一些例题，看一看容斥原理在信息学中的应用。

2.3.1 HAOI2008 硬币购物

问题 2.3.1. 有 4 种面值的硬币，第 i 种硬币的面值是 c_i。有 n 次询问，每个询问中第 i 种硬币的数目是 d_i，以及一个购物款 s，回答付款方法的数目。数据规模 $n \le 10^3, s \le 10^5$。

这题初一看是一个经典的多重背包问题，但是经过分析，我们发现单次动态规划的最好复杂度是 $O(4s)$，对于多次询问根本无法承受。

但是这题与一般的背包问题有一个明显的不同：硬币（即不同的物品）只有 4 种。而且，若每次购物没有硬币数目的限制，可以用一个动态规划预处理后 $O(1)$ 回答每组询问。

考虑一次询问，第 i 种硬币使用的数目是 x_i，那么需要满足 $c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 = s$，且 $x_i \le d_i$。我们发现，这与之前的不定方程非负整数解计数非常类似，只不过每个变量前有一个系数。

同样我们用容斥原理来处理这个问题，S_i 表示满足 $x_i \le d_i$ 的解的数目，$\overline{S_i}$ 表示满足 $x_i \ge d_i + 1$ 的解的数目，考虑若干 $\overline{S_i}$ 的交集，即一些硬币使用数有下

限，我们同样可以从 s 中减去下界和，问题变成了对于一个 s'，若硬币使用数目无限制，有多少种不同的付款方式。而这是一个经典的无限背包问题，可以预处理。

所以对每组询问进行容斥，设最大的 s 为 m，那么总的时间复杂度就是 $O(4m + n \cdot 2^4)$。

考虑我们的解题过程，我们首先发现问题的经典算法时间复杂度过高，但是我们抓住了题目的特殊性，通过写出问题的数学形式，通过联想，应用容斥原理把问题拆分，减少了局部问题的限制数，最终解决了问题。

2.3.2 原创题 游戏

问题 2.3.2. Alice 和 Bob 在玩游戏。他们有一个 n 个点的无向完全图，设所有的边组成了集合 E，于是他们想取遍 E 的所有非空子集，对某个集合 S 有一个估价 $f(S)$，这个估价是这样计算的：考虑 n 个点与 S 中的边组成的图，我们用 m 种颜色对所有点染色，其中同一个联通块的点必须染成一种颜色，那么 $f(S)$ 等于这个图的染色方案数。同时，Alice 喜欢奇数，所以当 $|S|$ 为奇数时，Alice 的分值加上 $f(S)$，否则 Bob 的分值加上 $f(S)$。求最后 Alice 的分值减去 Bob 的分值的值模 10^9+7 的结果。数据规模 $n, m \le 10^6$。

显然我们无法枚举 E 的所有非空子集；另一方面，对于相同的 $|S|$，联通块数目也不尽相同。我们似乎找不到一个突破口。这种情况下，我们就应该写出问题的数学形式，再进行分析。

首先，一个事实是，“同一联通块必须染相同的颜色”与“有边直接相连的两点必须染相同的颜色”是等价的。于是我们可以对每个点设一个变量，用 x_i 表示第 i 个点的颜色，x_i 是 $[1, m]$ 中的整数，那么一条无向边 (i, j) 就表示一个等式 $x_i = x_j$。我们考虑 Alice 的得分 $scoreA$ 和 Bob 的得分 $scoreB$，令 $F(C)$ 表示在情况 C 下的染色数，用 $[C]$ 表示一个情况 C，则

$$\begin{aligned} scoreA &= \sum_{\varnothing \neq S \subseteq E, |S| \text{是奇数}} F\left(\bigcap_{(i,j) \in S} [x_i = x_j]\right) \\ scoreB &= \sum_{\varnothing \neq S \subseteq E, |S| \text{是偶数}} F\left(\bigcap_{(i,j) \in S} [x_i = x_j]\right) \end{aligned}$$

现在考虑 $ans = scoreA - scoreB$，即 $|S|$ 为奇数时贡献为正，$|S|$ 为偶数是贡献为负，容易想到加一个 -1 的幂将式子统一：

$$ans = \sum_{\varnothing \neq S \subseteq E} (-1)^{|S|-1} F\left(\bigcap_{(i,j)\in S} [x_i = x_j]\right)$$

我们把 $[x_i = x_j]$ 这 $\frac{n(n+1)}{2}$ 个情况用 P_i 代替，令 $t = \frac{n(n+1)}{2}$，则 P_i 的 i 的取值范围是 $1 \le i \le t$。令 $Q = P_i$，那么再考虑上式：

$$\begin{aligned} ans &= \sum_{\varnothing \neq S \subseteq Q} (-1)^{|S|-1} F\left(\bigcap_{P_i \in S} P_i\right) \\ &= \sum_i F(P_i) - \sum_{i<j} F(P_i \cap P_j) + \cdots + (-1)^{t-1} F(P_1 \cap P_2 \cap \cdots \cap P_t) \end{aligned}$$

注意到这个形式与容斥原理极其相似！我们可以根据容斥原理，逆向分析出上式右边所求值的含义，即

$$ans = F\left(\bigcup_{i=1}^{t} P_i\right)$$

考虑上式右边的含义，即**至少有两个点颜色相同**的染色数！那么该问题中全集是点的染色方案集合，通过补集转化，我们就只需要求**点两两颜色不同**的染色数！而这个的计算方法是显然的，答案是 $\prod_{i=1}^{n} (m - i + 1)$。所以原问题答案就是 $m^n - \prod_{i=1}^{n}(m - i + 1)$。

细心的同学应该发现了，上面的式子中存在一个函数 F，它对一个情况，即一些条件的交定义，其实我们考虑满足 P_i 的染色方案构成集合 S_i，那么其实 $F\left(\bigcap P_i\right) = |\bigcap S_i|$，这样就和之前的容斥原理形式一致了。

回顾我们的解题过程，我们首先直接写出了答案的数学形式，把一些文字条件转化为数学条件，再进行一些换元、代入，得到一个关于若干条件的交集的式子，最终得到容斥原理的形式，**逆向分析**出问题的本质，找出算法并解决问题。如果说原来的容斥原理都是通过着眼局部，整合答案，在某种意义上进行了“微分”，那么这道题目中我们就是用的整体分析的方法，对答案的一个冗长的式子进行了“积分”，得到一个简洁的答案。这两个方向都体现了信息学中的**转化**思想。

3 容斥原理的推广

3.1 数论中的容斥原理

我们考虑一个经典的问题：给一个正整数 n，求 1 到 n 中与 n 互质的数的个数 $\varphi(n)$。

事实上我们要求的是 $|\{x|1 \le x \le n, gcd(x,n) = 1\}|$，其中 $gcd(a,b)$ 表示 a 和 b 的最大公约数。注意到这也是一个对某个集合计数的问题，但是 $gcd(x,n) = 1$ 这个限制太“大”，因为 gcd 这个函数本身较“复杂”，所以，我们应该想到从最大公约数的性质，去把 $gcd(x,n) = 1$ 这个限制拆成若干个小的限制。

我们考虑两个数 a 和 b 的质因数分解，若 $a = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$，$b = p_1^{b_1} p_2^{b_2} \cdots p_k^{b_k}$，那么我们有 $gcd(a,b) = p_1^{min(a_1,b_1)} p_2^{min(a_2,b_2)} \cdots p_k^{min(a_k,b_k)}$，其中 $min(x,y)$ 表示 x 和 y 中的较小值。

注意到，若两个数 a 和 b 的最大公约数是 1，那么它们的因数分解中一定没有相同的质数，而这是一个充要条件！所以，若 n 的不同的质因子有 $p_1, p_2, \cdots, p_k$ 共 k 个，那么我们需要统计的 x 就要同时满足 k 个条件，即对于 $1 \le i \le k$，都有 x 不是 p_i 的倍数。

现在我们可以把我们的结论写成数学的形式。设 P_i 表示 x 不是 p_i 的倍数这个性质，S_i 表示 1 到 n 中满足 P_i 的数组成的集合，那么这里的全集 U 就是 1 到 n 的整数集合，我们需要求的就是：

$$
\begin{aligned}
|\{x|1 \le x \le n, gcd(i,n) = 1\}| &= \left|\bigcap_{i=1}^{k} S_i\right| \\
&= |U| - \left|\bigcup_{i=1}^{k} \overline{S_i}\right| \\
&= n - \left|\bigcup_{i=1}^{k} \overline{S_i}\right|
\end{aligned}
$$

这就是一个容斥原理的式子！

再考虑 $\cap_i \overline{S_i}$ 的含义，它表示的是对于一些质数，我们统计 $[1,n]$ 上有多少个数同时是这些数的倍数。这个的统计方法非常简单：设质数的积为 m，那么答案就是 $\frac{n}{m}$。

所以我们可以写出我们所求答案的表达式：

$$\begin{aligned}|\{x|1 \le x \le n, gcd(i,n)=1\}| &= n - \sum_i \frac{n}{p_i} + \sum_{i<j} \frac{n}{p_i p_j} - \cdots + (-1)^k \frac{n}{p_1 p_2 \cdots p_k} \\ &= n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\cdots\left(1-\frac{1}{p_k}\right) \\ &= n\prod_{i=1}^{k}\left(1-\frac{1}{p_i}\right)\end{aligned}$$

这其实就是著名的欧拉公式。

从这个例子可以发现，我们容斥时考虑的是一个质数的集合，而我们取遍这个集合的子集时，得到的质数的乘积中所有质因子的次数都是 1，我们称这样的数为**无平方因子数**。再看 1 到 n 中每个 n 的约数对答案的贡献，显然只有 1 和无平方因子数有贡献，而且无平方因子数所作贡献的正负与质因子的个数有关。定义一个函数 $\mu(n)$，它定义在正整数集合上，且

$$\mu(n) = \begin{cases} 1 & n = 1 \\ (-1)^k & n = p_1 p_2 \cdots p_k \\ 0 & \text{otherwise} \end{cases}$$

这其实就是著名的**莫比乌斯函数**。我们再重新考虑之前的问题，容斥过程中的表达式可以写成 $\varphi(n) = \sum_{d|n} \mu(d) \frac{n}{d}$。

数论中的很多计数问题都可以用类似的方法解决：考察“最小元”即质数，计算“部分”即每个约数对答案的贡献，利用莫比乌斯函数进行容斥。在数论中，还有一种方法叫**莫比乌斯反演**，有兴趣的同学可以另行探究。

3.2 概率论中的容斥原理

在概率论中，对于一个概率空间内的 n 的事件 $A_1, A_2, \cdots, A_n$，也存在着一个容斥原理：

$$\mathbb{P}\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1} \mathbb{P}(A_i) - \sum_{i<j} \mathbb{P}(A_i \cap A_j) + \cdots + (-1)^{n-1} \mathbb{P}(A_1 \cap A_2 \cap \cdots \cap A_n)$$

若事件的交集发生的概率只和事件的数量有关，且设 k 个事件的交集的概率为 a_k，那么可以用组合数简化：

$$\mathbb{P}\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{k=1}^{n} (-1)^{k-1} C_n^k a_k$$

容斥原理在概率论中的实际应用比较少见，笔者也没有用容斥原理解决概率问题的经验。这个领域仍需更深一步的探究。

4 容斥原理的一般化

4.1 预备知识

由前面可知，容斥原理适用于对集合的计数问题，其实，对于两个关于集合的函数 $f(S)$ 和 $g(S)$，若

$$f(S)=\sum_{T\subseteq S}g(T)$$

那么就有

$$g(S)=\sum_{T\subseteq S}(-1)^{|S|-|T|}f(T)$$

这是一个更加一般的形式，而且对于之前讨论过的几种情形下的容斥原理都能找到 $f(S)$ 和 $g(S)$ 函数进行对应，其中 S 表示的是 n 个性质的集合。由于找到的 $f(S)$ 和 $g(S)$ 形式很复杂，在此略过，有兴趣的同学可以参考维基百科“容斥原理”词条。

另外，上面的式子也可以稍加变形写成这样：

$$\begin{aligned}f(S)&=\sum_{T\supseteq S}g(T)\\ g(S)&=\sum_{T\supseteq S}(-1)^{|S|-|T|}f(T)\end{aligned}$$

其实只用把之前式子中 S 和 T 替换成关于全集的补集，$\subseteq$ 号就换成 $\supseteq$ 了。

下面我们通过一个例子来感知一下。

4.2 例题：有标号 DAG 计数

问题 4.2.1. 给出 n，对 n 个点的有标号有向无环图进行计数，输出答案模 10^9+7 的值。数据规模 $n\leq 5\times 10^3$。

这是一类图的计数的问题。我们考虑动态规划，因为有向无环图中有一类特殊的点，即 0 入度的点，所以记 $dp(i,j)$ 表示 i 个点的有向无环图，其中恰有

j 个点的入度为 0，的答案，那么我们考虑去掉这 j 个点后，还有 k 个点入度为 0，写出转移

$$dp(i,j) = C_i^j \sum_{k=1}^{i-j} (2^j-1)^k 2^{j(i-j-k)} dp(i-j,k)$$

C_i^j 表示从 i 个点中选出 j 个点的选法，而去掉 j 个点后的 k 个 0 入度点与这 j 个点间至少有 1 条边即 $(2^j-1)^k$，然后这 j 个点还可以往除了这 $j+k$ 个点之外的点随意连边即 $2^{j(i-j-k)}$。这个算法时间复杂度 $O(n^3)$。

注意我们在定义状态时，是“0 入度点恰好为 k”，因为限制过严，导致我们需要考虑的很多。一个常见的办法是，在状态定义中将“恰好”改成“不少于”以放宽限制。但在这个问题中，从 i 个点选不少于 j 个 0 度数点，选法很多，转移时重复计算的情况很复杂，我们可以考虑将这不少于 j 个的点特殊化，即

我们记 $f(n,S)$ 表示 n 个点，只有 S 中的点的入度为 0；类似地定义 $g(n,S)$ 表示 n 个点，至少 S 中的点的入度为 0。可以发现 $g(n,S)$ 的转移比较简单：

$$g(n,S) = 2^{|S|(n-|S|)} g(n-|S|,\varnothing) \tag{1}$$

另一方面，我们再考虑 $f(n,S)$ 和 $g(n,S)$ 的关系，这也比较简单：

$$g(n,S) = \sum_{T\supseteq S} f(n,T) \tag{2}$$

注意式子(2)与之前提到的一般化的容斥原理相似，不妨将之应用：

$$f(n,S) = \sum_{T\supseteq S} (-1)^{|S|-|T|} g(n,T) \tag{3}$$

而我们的目的是求 $g(n,\varnothing)$，先使用式子(2)进行推导：

$$\begin{aligned} g(n,\varnothing) &= \sum_{\varnothing\neq T} f(n,T) \\ &= \sum_{m=1}^{n} \sum_{|T|=m} f(n,T) \end{aligned}$$

再代入我们用容斥原理推出的式子(3)：

$$\begin{aligned} g(n,\varnothing) &= \sum_{m=1}^{n}\sum_{|T|=m}\sum_{S\supseteq T}(-1)^{|T|-|S|}g(n,S) \\ &= \sum_{m=1}^{n}\sum_{|T|=m}\sum_{S\supseteq T}(-1)^{|T|-|S|}2^{|S|(n-|S|)}g(n-|S|,\varnothing) \\ &= \sum_{m=1}^{n}\sum_{|T|=m}\sum_{k=m}^{n}C_{n-m}^{k-m}(-1)^{k-m}2^{k(n-k)}g(n-k,\varnothing) \\ &= \sum_{m=1}^{n}C_n^m\sum_{k=m}^{n}C_{n-m}^{k-m}(-1)^{k-m}2^{k(n-k)}g(n-k,\varnothing) \end{aligned}$$

利用一些组合数的性质可以继续进行化简。这里直接给出最后的化简结果：

$$g(n,\varnothing)=\sum_{k=1}^{n}(-1)^{k-1}C_n^k 2^{k(n-k)}g(n-k,\varnothing)$$

注意到此时我们计算 $g(n,\varnothing)$ 的时间复杂度降到了 $O(n^2)$，容斥原理在中间起到了举足轻重的作用。

回顾我们的解题过程，首先我们在定义状态时放宽了状态的限制，这样可以认为新的状态是之前状态某种意义下的“前缀和”，列出等式后用容斥原理得到另一个式子，然后整合我们手中的等式推导答案的表达式，最后得到复杂度较低的算法。

5 总结

容斥原理是组合数学中一个重要的定理，在解决问题的时候，我们既可以使用“隔离法”，将所需求的解要满足的条件拆分，放宽限制，解决若干简单的子问题，再整合答案；也可以使用“整体法”，对所求的式子进行整体感知，逆向地合并条件，找出问题的本质。这里体现了**转化**的思想，当然在思考过程中也需要一些数学功底。

容斥原理同时并不是仅仅应用于组合计数，稍加变形后就可以解决一些数论或概率论的问题，其思想是一致的。而最后我们通过一些资料得知了容斥原理更为一般的形式，它适用于定义在集合上的函数，这使得容斥原理更加抽象，也让我们开阔了思路，即在一些情况下，我们用集合的形式描述我们的算法，利用容斥原理得到另外的等式，这相当于增加了已知量，使得问题更容易入手。

在研究过程中，我从解决计数问题中体会到了一些信息学中的思维方法：转化、特殊化（放宽限制）、逆向分析，开阔了眼界；同时，在查阅容斥原理相关资料的过程中，意识到了平时学习的各种算法，其背后或许仍有继续研究的空间，所以我们应不断求知，将学习到的知识有机整合，并思考它们的本质，体会不同的算法后面的思想，形成自己的知识网络，增强自己的思维能力。

参考文献

1. http://en.wikipedia.org/ 维基百科
2. http://www.math.ust.hk/~mabfchen/Math232/Inclusion-Exclusion.pdf
3. 顾昱洲，《Graphical Enumeration》

感谢

- 感谢父母对我的养育
- 感谢我的教练成都七中的张君亮老师，以及其他所有给予我支持的老师
- 感谢罗雨屏、李凌霄、钟皓曦等同学的帮助
- 感谢 CCF 给我一个展示自己的机会