「学习总结」容斥原理

Jiayi Su (ShuYuMo)

2021-01-24 17:29:33

容斥原理 用于解决一类，在已知任意 m 个集合交集大小的情况下，多个集合求并集大小的问题。

容斥原理

定义及证明

设 U 中元素有 n 种不同的属性，而第 i 种属性称为 P_i ，拥有属性 P_i 的元素构成集合 S_i ，那么

$$\left|\bigcup_{i=1}^{n} S_i\right| = \sum_{m=1}^{n}(-1)^{m-1}\sum_{a_i<a_{i+1}}\left|\bigcap_{i=1}^{m} S_{a_i}\right|$$

其中 a 为任意长度为 m 且 值域为 $[1,n]$ 的不重无序数列。

通过定义可知，容斥原理 用于解决一类，在已知任意 m 个集合交集大小的情况下，多个集合求并集大小的问题。

对于有限制条件的计数问题，可以转化成求集合交并大小问题，进而通过容斥原理解决。

关于容斥原理的证明，其实就是要保证并集中的每一个元素对答案的贡献为 1 。

对于元素 x，假设它出现在 $T_1, T_2, \cdots, T_m$ 的集合中，那么它的出现次数为

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

计数问题的转化

可以考虑把具有相同属性的计数对象放入同一集合。然后根据题目要求，求出同时具有某些属性的技术对象个数（即：属性对应的集合交集）。

$$\left|\bigcap_{i=1}^{n} S_i\right| = |U| - \left|\bigcup_{i=1}^{n} \overline{S_i}\right|$$

由于容斥原理本身是求集合并集大小，但是可以通过 上式 转化为求集合交集大小问题。

容斥原理栗题（们）

栗题一

详见不定方程非负整数解计数。

对于限制元素数量下界的要求，处理方式都可以采用，直接对总数减去这个元素的下界，计算取值时直接不考虑下界即可。

栗题二

对于一个 1~n 的排列 P，若 $\forall i, P_i \neq i$ 则称其为错位排列。给出 n ，求长度为 n 的错位排列个数。

考虑全集 $|\mathbb{U}| = n!$ ，元素属性就是 $P_i \neq i$，答案就是 $\left|\bigcap\limits_{i=1}^{n} S_i\right|$。

考虑到：

$$\left|\bigcap_{i=1}^{n} S_i\right| = |\mathbb{U}| - \left|\bigcup_{i=1}^{n} \overline{S_i}\right|$$

易知：$\overline{S_i}$ 就是所有 $P_i = i$ 的排列。

考虑到：

$$\begin{aligned}\left|\bigcup_{i=1}^{n} \overline{S_i}\right| &= \sum_{k=1}^{n} (-1)^{k-1} \sum_{a_1,\ldots,a_k} \left|\bigcap_{i=1}^{k} S_{a_i}\right| \\ &= \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} (n-k)!\end{aligned}$$

综合上式，得出长度为 n 的错位排列数为：

$$D_n = n! - \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} (n-k)!$$

栗题三

A 和 B 喜欢对图（不一定连通）进行染色，而他们的规则是，相邻的结点必须染同一种颜色。

今天 A 和 B 玩游戏，对于 n 阶 完全图 $G = (V, E)$ 。他们定义一个估价函数 $F(S)$ ，其中 S 是边集，$S \subseteq E$.

$F(S)$ 的值是对图 $G' = (V, S)$ 用 m 种颜色染色的总方案数。

他们的另一个规则是，如果 $|S|$ 是奇数，那么 A 的得分增加 $F(S)$ ，否则 B 的得分增加 $F(S)$. 问 A 和 B 的得分差值。

~~出题人千辛万苦凑出的式子~~

考虑形式化的定义答案：

$$Ans = \sum_{S \subseteq E} (-1)^{|S|-1} F(S)$$

设集合 $Q_{(i,j)}$ 中的元素为所有 (i, j) 有边相连的图的染色方案。考虑到相邻的节点（有边相连）必须染成相同的颜色，所以两节点 i, j 有边相连即 节点 i, j 染成相同的颜色。

易知：

$$F(S) = \left|\bigcap_{e \in S} Q_e\right|$$

带入原式即：(这里用到了容斥原理的逆用)

$$
\begin{aligned}
Ans &= \sum_{S \subseteq E} (-1)^{|S|-1} \left| \bigcap_{e \in S} Q_e \right| \\
&= \left| \bigcup_{e \in E} Q_e \right|
\end{aligned}
$$

答案变成：对一张完全图染色，存在任意两个点同色的方案数。

考虑到两两点都异色的染色方案数为 A_m^n。

所以答案为：

$$m^n - A_m^n$$

其实容斥原理本质上是集合间交集和并集大小之间的转化。

栗题四

求出：

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \gcd(i, j)$$

其中：$n \le 10^5$

考虑枚举 gcd ，设函数 $f(g)$ 为 以 g 为最大公约数的数对个数。易知：

$$f(g) = \lfloor \frac{n}{g} \rfloor^2 - \sum_{i=2}^{i \times g \le n} f(i \times g)$$

考虑到当 $g > \frac{n}{2}$ 时，可以直接得到答案。其余的值逆向递推即可。### 栗题五 容斥原理推导欧拉函数通项公式

栗题六

询问 $1-n$ 中有多少数字可以表示成 $x^y, y > 1$ 的形式。其中 $n \le 10^{18}$

枚举 x 的复杂度为 $\mathcal{O}(\sqrt{n})$ 的。考虑枚举 y ，这样的复杂度仅为 $\mathcal{O}(\log n)$。枚举一个 y 后，合法的数字有 $\sqrt[y]{(n)}$ 个。

易知，当 y 不等于质数积时，贡献为 0。例如 $y = 4$ 时，这里的答案一定被 $y = 2$ 时算过一次了。

其余的情况，根据容斥原理的套路，可以发现，容斥系数为 $-\mu(y)$ 。莫比乌斯函数也被称之为数论容斥系数。

栗题七

DAG 计数。给出点数 n ，输出 n 个点的带标号 DAG 的数量，对大质数取模。其中 $n \le 5 \times 10^3$

考虑到对于一个 DAG 来说，将其入度为 0 的点剖去之后，剩下的图也是一个 DAG 。这样就成功划分了子问题。

朴素做法　设 $f(i, j)$ 表示 i 个点的 DAG，有 j 个点的入度为 0，考虑转移：枚举剥去这 j 个点后会剩下多少个入度为 0 的点。

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

后面的式子分别为：

- $\binom{i}{j}$ ：在 i 个标号中选出 j 个充当入度为 0 的点。
- $(2^j - 1)^k$：对于这 k 个入度为 0 的点，他们可以和之前的 j 个点随意连边（除了不连任何边的情况）。

- $2^{j(i-j-k)}$：对于这 j 个点，还可以与除这 k 个点剩下的 $i-j-k$ 个点任意连边，一共有 $i\times(i-j-k)$ 条边可以连。这样的做法是 $\mathcal{O}(n^3)$ 的

优化做法 容斥原理的一般化：对于两个集合函数 $f(S),g(S)$：

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

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

前面的式子是 FMT 莫比乌斯变换所加速的式子。

设：$f(n,S)$ 为 n 个点的 DAG 中 S 中的点度数为 0，类似地，$g(n,S)$ 为 n 个点的 DAG 中 至少 S 中的点度数为 0（钦点）。易知：

$$g(n,S)=2^{|S|(n-|S|)}g(n,\emptyset)$$

其中 g,f 有如下关系。

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

根据容斥原理一般公式：

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

目的是求出 $g(n,\emptyset)$：

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

带入 g,f 的关系式：

$$
\begin{aligned}
g(n,\emptyset) &= \sum_{T\neq\emptyset} f(n,T) \\
&= \sum_{i=1}^{n} \sum_{T,|T|=i} f(n,T) \\
&= \sum_{i=1}^{n} \sum_{T,|T|=i} \sum_{S\supseteq T} (-1)^{|S|-|T|} g(n,S) \\
&= \sum_{i=1}^{n} \sum_{T,|T|=i} \sum_{S\supseteq T} (-1)^{|S|-|T|} 2^{|S|(n-|S|)} g(n-|S|,\emptyset) \\
&= \sum_{i=1}^{n} \sum_{T,|T|=i} \sum_{S\supseteq T} (-1)^{|S|-i} 2^{|S|(n-|S|)} g(n-|S|,\emptyset) \\
&= \sum_{i=1}^{n} \sum_{T,|T|=i} \sum_{k=i}^{n} \binom{n-i}{k-i} (-1)^{k-i} 2^{k(n-k)} g(n-k,\emptyset) \\
&= \sum_{i=1}^{n} \binom{n}{i} \sum_{k=i}^{n} \binom{n-i}{k-i} (-1)^{k-i} 2^{k(n-k)} g(n-k,\emptyset) \\
&= \sum_{k=1}^{n} 2^{k(n-k)} g(n-k,\emptyset) \sum_{i=1}^{k} \binom{n}{i} \binom{n-i}{k-i} (-1)^{k-i} \\
&= \sum_{k=1}^{n} 2^{k(n-k)} g(n-k,\emptyset) \sum_{i=1}^{k} \binom{n}{k} \binom{k}{i} (-1)^{k-i} \\
&= \sum_{k=1}^{n} 2^{k(n-k)} g(n-k,\emptyset) \binom{n}{k} \sum_{i=1}^{k} \binom{k}{i} (-1)^{k-i} \\
&= \sum_{k=1}^{n} 2^{k(n-k)} g(n-k,\emptyset) \binom{n}{k} \left[\left(\sum_{i=0}^{k} \binom{k}{i} (-1)^{k-i} 1^{i} \right) - (-1)^k \right] \\
&= \sum_{k=1}^{n} 2^{k(n-k)} g(n-k,\emptyset) \binom{n}{k} \left[(1-1)^k - (-1)^k \right] \\
&= \sum_{k=1}^{n} 2^{k(n-k)} \binom{n}{k} (-1)^{k-1} g(n-k,\emptyset)
\end{aligned}
$$

这样的做法是 $\mathcal{O}(n^2)$ 的。

扩展容斥原理

Min-max 容斥

$$\max S = \sum_{T\subseteq S} (-1)^{|T|-1} \min T$$

{#eq:max_min}

$$\min S = \sum_{T\subseteq S} (-1)^{|T|-1} \max T$$

「**PKUWC2018**」随机游走

给定一棵 n 个结点的树，你从点 x 出发，每次等概率随机选择一条与所在点相邻的边走过去。

有 Q 次询问，每次询问给定一个集合 S，求如果从 x 出发一直随机游走，直到点集 S 中所有点都至少经过一次的话，期望游走几步。

特别地，点 x（即起点）视为一开始就被经过了一次。

答案对 \$998244353 \$ 取模。

对于 100% 的数据，有 $1 \leq n \leq 18$，$1 \leq Q \leq 5000$，$1 \leq k \leq n$

设 A_i 表示到达从 x 点出发第一次到达点 i 的期望时间。

易知：答案就是 $E\left(\max\limits_{i \in S}(x_i)\right)$。需要注意的是：$E\left(\max\limits_{i \in S}(x_i)\right) \neq \max\limits_{i \in S}(E(x_i))$，详见博文

> 这是非常有用的，因为期望下的 max 和 min 是很难求的。
>
> 假设有 a, b 两个不相关变量，则 $E(\max(a,b)) \neq \max(E(a), E(b))$。
>
> 例子：抛硬币，$a = b = \begin{cases} 0(50\%) \\ 1(50\%) \end{cases}$，则 $E(a) = E(b) = \dfrac{1}{2}$
>
> 那么 $\max(a,b) = \begin{cases} \max(0,0)(25\%) \\ \max(0,1)(25\%) \\ \max(1,0)(25\%) \\ \max(1,1)(25\%) \end{cases}$，则 $E(\max(a,b)) = 0.75$
>
> 但是 $\max(E(a), E(b)) = 0.5$ 所以期望不能大力拆 max 或 min 。
>
> ——引用自 command_block 的博客。

由 (eq.~@eq:max_min) 可知：

$$E\left(\max_{i \in S} x_i\right) = \sum_{T \subseteq S} (-1)^{|T|-1} E\left(\min_{i \in T} x_i\right)$$

考虑如何求出 $E\left(\min\limits_{i \in T} x_i\right)$

相当于从点 x 出发，首次到达 T 中的任意一点的期望时间。

设 $f(i)$ 表示从结点 i 出发，到达 T 首次中的点的期望时间。

对于 $i \in T, f(i) = 0$

对于 $i \notin T, f(i) = \dfrac{1}{deg_i}\left(\sum\limits_{v}(f(v)+1) + f(fa_i) + 1\right)$

~~据说是~~经典套路：

待定系数法，设 $f(i) = A_i \times f(fa_i) + B_i$

$$f(i) = \frac{1}{deg_i}\left(\sum_{v}(f(v)+1) + f(fa_i) + 1\right)$$

$$= \frac{1}{deg_i}\left(\sum_{v}(A_v f(i) + B_v + 1) + f(fa_i) + 1\right)$$

$$= \frac{1}{deg_i - \sum A_v} f(fa_i) + \frac{deg_i + \sum B_v}{deg_i - \sum A_v}$$

所以：$A_i = \dfrac{1}{deg_i - \sum A_v}, B_i = \dfrac{deg_i + \sum B_v}{deg_i - \sum A_v}$

特殊的：对于 $i \in T$, $A_i = B_i = 0$。

这里的 A,B 可以直接通过树上 dp 求出。同时，可以递推出 $f(i)$ 的值。

设 $F(T)=(-1)^{|T|-1}f(x)$, 答案就是 $\sum\limits_{T\subseteq S}F(T)$

考虑到这东西是子集和变换，使用 FMT (快速莫比乌斯变换) 预处理即可。然后 $O(1)$ 回答。