prufer序列

prufer序列长度为n - 2，与有标号无根树一一对应

（1）无根树转化为prufer序列。

首先定义无根树中度数为1的节点是叶子节点。

找到编号最小的叶子并删除，序列中添加与之相连的节点编号，重复执行直到只剩下2个节点。

（2）prufer序列转化为无根树

维护两个集合，S是不在prufer序列中出现编号，T是在prufer序列中出现编号，每次取S的最小值，与序列当前值连边，然后S中去掉最小值，T中序列当前值次数-1，如果为0将其加入S中，最后S中剩下两个编号连边

（3）每个点在prufer序列中出现次数是du[i] - 1。

（4）有标号无根树数量为 n^{n-2}

（5）一些点度数确定时，相当于在n-2个位子中先选这些点的位子，其他位子值随便取其他值，所有点都确定是数量是

$$\frac{(n-2)!}{\prod_{i=1}^{n}(d_i-1)!}$$

（6）有标号有根树数量，每一种有标号无根树选一个标号为根都是不同的，所以数量为

$$n^{n-2}*n=n^{n-1}$$

（7）n个点，k个联通块求有标号无根树的方案数

假设我们有 n 个点，被分成了 k 个点集，每个点集里的点已经连通，不同点集之间的点两两无边，现在我们要在这个 n 个点 n-k 条边的基础上求生成树个数。设第 i 个点集包含的点数为 size[i] 。

那么，如果我们把这 k 个点集每一个点集都看作一个点，做一个 k 个点的生成树，那么有k^{k-2} 种方案；但是由于这里的每一个点都是一个点集，所以假设它是点集 i，那么从他连出去的每一条边的属于集合i的端点，都有 size[i] 种选法。也就是说，对于一个 k 个点的 prufer 编码，假设在这个编码中，数字 i 出现了 c[i] 次，那么这个编码对应到原树上就会贡献$\prod_{i=1}^{k} size[i]^{c[i]+1}$次。

我们把每一个 "c[i]+1" 中多出来的 1 提出，看作常量，我们来对于所有 prufer 编码求贡献总和：

$$\sum_{P是一个prufer编码}\prod_{i=1}^{k} size[i]^{c[i]}$$

考虑到这个prufer编码的每一位选择第 i 个点集，就会对乘积有 size[i] 的贡献，根据乘法分配律，我们可以得到上面的那个式子就是：

$$(\sum_{i=1}^{k} size[i])^{k-2}=n^{k-2}$$

再乘上之前提出的东西，所以答案就是：

$$(\sum_{i=1}^{k} size[i])^{k-2} \cdot \prod_{i=1}^{k} size[i]$$

这个可以推出比如n个点k个森林，1-k号点分别属于不同森林的数量，其实可以看成1-k属于一个联通块，其他点各自是一个联通块的计数

min-max容斥

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

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

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

主要用于一些求期望的题目，求一些随机变量的最大值可以转化为求这些随机变量最小值实现

kthmax容斥这个系数是可以关于k进行递推的，如洛谷4707

$$lcm(S) = \prod_{T \subset S} gcd(T)^{(-1)^{|T|+1}}$$

若有

$$g(n) = \prod_{T \subset 2^{[n]}} f(gcd_{i \in T}(i))^{(-1)^{|T|+1}}$$

可构造出h满足

$$f(n) = \prod_{d|n} h(d)$$

于是

$$g(n) = \prod_{d=1}^{n} h(d)$$

高维前缀和

其实就是对每一维做前缀和，这样对每一维做一遍以后就可以的到全部的前缀和了

对于一个二进制数，可以看成是一个n维向量，这样对每一维都做一遍相当于得到了每一个数的子集和

也可以统记固定一些位是1，其它位随便取的和，主要就是看是0转移到1还是1转移到0

遇到一道题，其实符合前缀性质的都可以这样做，可以维护前缀min，这样可以知道固定几位的那种数字最早什么时候出现。

也可以看成字典树上对每一位把0合并到1上