

## 结题报告

### Hdu 4336 Card Collector

这道题其实是我YY过的，看测试数据很像可以 $2^m$ 枚举，然后把枚举到的卡片概率加起来算期望，奇数加偶数减，然后就过了。想不清楚为什么可以用容斥原理，然后就找了个挺靠谱的解释自我安慰， $E1$ 表示买到1的期望, $E1 = 1/p1$ ，也就是说 $E1$ 包里面肯定包含1这张卡片，当我们计算 $E1$ 和 $E2$ 时，是不是会有一种情况：我们想要卡片1的时候已经买到了卡片2，然后我们又要计算买卡片2的期望，正是因为这样的交集使得我们可以用容斥，交集的期望 $E12 = 1 / (p1 + p2)$  表示肯定买到1、2中的其中一包， $E123$ 就表示肯定买到1、2、3中的某一种,我们在计算 $E12, E13, E23$ 的时候 $E123$ 多减了一次，要加回来，以此类推....

这题有状态压缩DP解法， $dp[i]$ 表示 $i$ 这个状态到最终状态需要的期望卡片数, $dp[i] = dp[i] * myself + \sum(dp[j] * p[k] = \sum(dp[j] * p[k]) / (1 - myself)$  ( $i|k=j$ 且 $i!=j$ , $myself$ 表示保持原状的概率).

### Hdu 1796 How many integers can you find

求 $[1, r]$ 内有因子出现在集合中的数的个数。题目要求的数要么含有一个集合中的因子，要么是三个..要么是 $m$ 个，而含两个集合中因子的数在计算含有一个集合中的因子，被重复计算了，三个时候也在计算二个的时候被重复计算，所以需要用到容斥原理。 $2^m$ 枚举集合中元素，然后计算出最小公倍数， $n/LCM$ 就是 $1..n$ 中含我们枚举的因子的数的个数对于求出来的数根据枚举到素因子个数奇数加偶数减。

### Ural 1091. Tmutarakan Exams

这题凑够正面入手。从 $[1, m]$ 中找 $n$ 个数，使得这些数的最大公约数大于1.因为 $m$ 非常小，我们可以把 $[2, m]$ 内的所有质因子都找出来，总数不超过15个，然后 $2^{cnt}$ 枚举这些质因子，求最小的LCM，求出 $[2, m]$ 内有几个数可以被LCM整除设为 $k$ 个，这次选到的集合即为 $C[k][n]$ ，然后判断选到的质数的个数，奇数加偶数减。

这题 $m$ 如果变成200，就不能这样做了，200的时候公因子有46个，枚举的话时间复杂度不允许。可以用DP来做， $dp[i][j][k]$ 表示选了 $i$ 个数，选到 $j$ 这个数，最大公约数为 $k$ 的方案数，最后输出 $dp[n][j][k](j, k = 2...m)$ ，复杂度 $O(n * m * m * \sqrt{m} * \log(m))$ ,这个复杂度看上去很可怕，但实际运算量并不大。

### Zoj 2836 Number Puzzle

和上题差不多，但要注意上界，然后要用long long，多敲加深对容斥原理地理解

### Hdu 4135 Co-prime

求 $[l, r]$ 内与 $x$ 互素的个数。这种以某个区间为背景的题目往往是 $\text{fun}(r) - \text{fun}(l-1)$ ,  $\text{fun}(x)$ 函数是求 $[1, r]$ 与 $x$ 互素的数的个数。求互素个数的时候我们从反面求解, 即求与 $x$ 不互素的数的个数即与 $x$ 有某些共同因子的数的个数即与 $x$ 有些素因子的数的个数, 先将 $x$ 的素因子全部找出来, 个数设为 $n$ ,  $2^n$ 枚举素因子, 求最小公倍数 LCM,  $r/\text{LCM}$ 就是 $1\dots r$ 中我们要求的不互素的个数, 对于求出来的数根据枚举到素因子个数奇数加偶数减。

### Hdu 4085 The Boss on Mars

找 $[1, n]$ 与 $n$ 互素的数为 $x$ 集合, 求 $\sum(i^4) - \sum(x_i^4) (1 \leq i \leq n)$ 。仍然找有相同因子的数, 先求质因子, 然后 $2^m$ 去枚举, 然后套个 $1^4+2^4+3^4\dots$ 的公式求和再乘以枚举到的所有质因子的 LCM 的 4 次方, 最后用总和去减求得和即可。四次方求和公式为  $\text{sum}n = n(n+1)(2n+1)(3n^2+3n-1)/30$ 。因为要除以 30, 所以要求逆元。

### Poj 2773 Happy 2006

容斥原理+二分。因为要查找第 $k$ 个与 $n$ 互质的数, 直接查找无从下手。这里有个重要的性质,  $[1, m+1]$ 内与 $n$ 互质的数的数量肯定不小于 $[1, m]$ 内与 $n$ 互质的数的数量, 也就是具备单调性, 那么我们二分枚举这个  $m$ , 算出 $[1, m]$ 有几个数与 $n$ 互质极为 $\text{cnt}$ , if ( $\text{cnt} \geq k$ )  $\text{high} = \text{mid} - 1$ , else  $\text{low} = \text{mid} + 1$ 。最后的答案就是 $\text{low}$ 。

### Uva 4683 Find The Number

容斥原理+二分。给定一个大小元素小于 12 的集合, 求第 $k$ 个注册号, 注册号是指被且只被 1 个集合中的元素整除的数。和上题一样, 二分枚举区间 $[1, m]$ 的 $m$ , 然后计算区间内有几个注册号, 计算的时候需要用到容斥原理。这里和普通的能整除的容斥不一样, 主要要求只被 1 个数整除, 那么如果某个数能被两个数整除, 那么相当于多加了 2 次, 如果能被三个数整除, 那么在去除 2 个数整除的数时多减了 3 ( $C(3, 2)$ ) 次...以此类推, 即可得解。

这题的题目描述有问题, 题面说答案只小等于 1015, 我想这么小, 那么  $O(1015 \cdot n)$  去计算每个数以前有几个符合情况不就好了? 果断 Wa 了, 其实 1015 的意思是  $10^{15}$ 。

### Hdu 2841 Visible Trees

求从  $(0, 0)$  能看到的二元组  $(x, y)$  数量, 一些树不能被看到是因为被与他成比例且比这个二元组小的树挡住。如果每个  $y$  都与  $x$  互质, 那么这些二元组里面绝对不会有成比例的二元组存在, 这样问题就变成了求 2 到  $n$  之间每个数  $i$  的小等于  $m$  的互质数个数总和。然后这个二元组  $(x, y)$  其实是和  $(y, x)$  等价, 那么我们强制性让  $x$  所

属的域小些。然后就可以用文章开头的那篇文章里倒二个例子的方法求解。

### Poj 3904 Sky Code

求从n个数里选择4个数使得这些数的最大公约数为1, 求四元组的数量。从方面思考, 计算最大公约数不是1的四元组数量, 用质因子的倍数组成的四元组数量减去两个质因子乘积的倍数组成的四元组数量加上... 然后用总的四元组数量去减上面的计算结果就是我们要的答案。计算方法和上题相似。

### Hdu 4407 Sum

这题初看之下没什么思路, 不能预处理, 容斥也无从下手, 而暴力模拟算法复杂度又不靠谱, 但是注意到m只有1000之后就好办了。每碰到一次询问, 就用容斥原理算一次[l,r]之间与p互质的数的总和, 然后再看下当前询问以前那些位置的数经过修改, 相应地调整下就可以AC了。而这里求与互质的数总和和求个数其实相差不多, 我们可以通过求和公式来算和, 其他和求个数一样。

### Ural 1114 Boxes

容斥原理+DP。用红球和白球填充n个一字排开的盒子, 每个盒子可以不放也可以放一种颜色的球或者两种颜色的球都放。比较容易用 $dp[i][j][k]$ 来表示状态,  $dp[i][j][k]$ 表示到第i个球放了j个红球k个白球的总方案数,  $dp[i][j][k] = dp[i-1][j][k] + dp[i-1][j-1][k] + dp[i-1][j][k-1] - dp[i-1][j-1][k-1]$ , 表示可以放0种颜色, 放1种颜色和2种颜色都放, 因为2种颜色都放在放一种的时候重复计算要减去这一部分。用unsigned long long就可以过, 无须高精度, 复杂度也是 $O(n*m*m)$ 。

网上还有种二维的dp写法, 把红球和篮球看成一样的球,  $dp[i][j]$ 表示到第i个盒子放了j个球的方案数,  $dp[i][j] = 1 + \sum(dp[k][j-1]) (1 \leq k \leq i)$ , 那个1表示j以前的球都不放。这种写法其实是上一种的优化, 本质是一样的, 复杂度也是 $O(n*m*m)$ 。

这题的方法如果改变下, 变成每个盒子放0个或1个或者多个各种颜色的球, 那么要怎么DP? 怎么容斥呢?

### Hdu 2461 (poj 3695) Rectangles

容斥原理+几何。多解, 但是用容斥原理来做这题速度最快。每个询问的面积=1个矩形面积-2个矩形交面积+3个矩形交面积..., 做法是去枚举矩形的集合, 然后判断这个集合在哪个询问里出现过, 然后看集合里的矩形个数, 奇数加偶数减。这种做法看上去很不错, 但其实很不靠谱, 你用普通的for循环来写就超时, 因为复杂度是 $O(2^n * m)$ , 很可怕的复杂度了。但是用深搜形式的我们可以来个强力剪枝, 如果到达某个状

态, 这里面的矩形交面积为0, 那么我们就可以不往下搜了。就这样水过了, 网上还有其他做法似乎更不靠谱...

### Zoj 3556 How Many Sets I

从方面思考, 用总的方案数减去非空的方案数。集合总数为 $2^n$ , 总的组合数为 $(2^n)^k = 2^{nk}$ . 非空的包含 $n$ 个数种的某个 $x$ 的集合数为 $p_1 = (2^{n-1})^k = 2^{(n-1)k}$ , 总的组合数是 $C(n, 1) * p_1$ , 我们在计算字少包含一个的时候重复计算包含了2个的情况, 那么同理 $p_2 = 2^{(n-2)k}$ , 组合数为 $C(n, 2) * p_2$ ...

最后得到一个等式 $ans = 2^{n*k} - C(n, 1) * p_1 + C(n, 2) * p_2 - \dots + (-1)^n * C(n, n) * p_n$ , 这个公式就是 $(2^k - 1)^n$ 的展开式。

上面的展开式并不好想, 最容易想到的思路是: 这个 $n$ 个数的任意一个只有在 $k$ 次选择中全选到最后交的时候才会出现, 那么这 $n$ 个其实是独立存在的, 分析单个数 $x$ 在 $k$ 次选择中全部情况有 $2^k$ 种, 这里面有一种选择最终会使得 $x$ 出现, 那么就有 $2^k - 1$ 种方案使得最后不出现 $x$ ,  $n$ 个数独立即方案数为 $(2^k - 1)^n$ 种。