一些计数姿势

visit_world

自我介绍?

- 吕欣，常用ID：visit_world
- (曾经) 活跃于 BZOJ、51nod
- 现在只是个菜鸡

先说两句?

- 这份课件的内容非常杂
- 不过都是一些(大家早就知道的)(没什么用的)小技巧
- 已经烂熟于心的同学可以放心睡觉啦

Contents

- 容斥原理
- DP技巧
- 图计数
- 数学Trick
- 杂题

PART 1: 容斥原理

- 介绍容斥原理更多的是要强调这种“以退为进”的计数思想
- 这一部分的内容比较简单，非常可听
- 当然也可能因为太简单所以有催眠的效果

容斥原理

- 一般模型：
  - 给你一堆 XXX 条件，问全部满足的方案数
- 满足若干条件的方案数很难算
- 但是它的反面，若干条件一定不满足的比较好算
- 于是使用容斥

容斥原理

- 给定 n 个条件 A1、A2……An，问全部满足的对象的个数
- 答案 = 所有对象 - 至少不满足其中一个的 + 至少不满足其中两个的 - 至少不满足其中三个的……
- 证明比较简单，考虑一个特定的、合法或不合法的对象会对上式产生多少贡献即可。

容斥原理

- 这里介绍两种在数学推导里常用的容斥形式
- 集合形式：
  - f[S] = ∑{T ⊆ S} g[T] —>
  - g[S]= ∑{T ⊆ S} (-1)^(|S|-|T|) * f[T]
- 限制条件高度对称：
  - f[i] = ∑{k=0}^{i} g[k] * C(i, k) —>
  - g[i] = ∑{k=0}^{i} (-1)^(i-k) * f[k] * C(i, k)

例1

- 给定 n, k 和大小为 k 的数组 {a}, {b}
- 求满足以下条件的大小为 k 的数组{c}的方案数：
  - $\sum c[i] = n$
  - $a[i] <= c[i] <= b[i]$ for each $i \in [0, n)$
- $n <= 10^9, k <= 20$

例1

- 没有上下界的时候可以插板法统计
- 下界限制比较容易搞，强行减掉就行
- 上界限制通过容斥转化为下界限制

例2

- 给定大小为 n 的数组 {a}
- 要求划分成非空的两组
- 使得两组中元素的 or 和相同
- n <= 50, a[i] < 2^20

例2

- 考虑每个二进制位
- 如果至少有一个数在这一位是 1，那么所有的 1 一定不能在同一组
- 枚举哪些位一定不满足条件，用并查集维护

例3

- 错位排列问题
- 求长度为 n 的排列 {a} 的个数，其中，a[i] != i
- n <= 10^5

例3

- 容斥做法：枚举有多少个位置有 a[i] = i
- 递推做法：f[i] = (i - 1) (f[i - 1] + f[i - 2])

例4

- 求 n 个点的图里的哈密顿路径的个数
- $n <= 18$

例4

- O(2^n * n) 的做法比较显然
- ......如果丧病出题人卡空间呢?
- 考虑路径构成哈密顿路的充要条件:
  - 路径共有 n-1 条边
  - 经过了 n 个不同的点
- 其实可以对条件二进行容斥，枚举哪些点一定没有经过，然后用一个 DP 统计出这种情况下，长度为 n-1 的路径个数

训练

- 非常简单
- 可能大家都见过
- ~~请假装不知道~~

BZOJ 4710

- 有 m 种特产，第 i 种有 a[i] 个
- 有 n 个同学分特产，要求：
  - 恰好分完
  - 每个人至少要分到一个
- 问方案数
- n, m, a[i] <= 1000

BZOJ 4710

- 裸题
- 枚举多少同学没有分到

BZOJ 4455

- 一个大小为 n 的图中嵌入一个大小为 n 的树的方案数
- n <= 18

BZOJ 4455

- 树的点到图的点是双射
- 双射等价于图中 n 个点都被映射到
- 枚举哪些点没有被映射，就能容斥了
- O(2^n * n^3)

SRM555: Hard

- 已知有一个初始值均未知的长度为 n 的 01 图灵机和一个磁头,定义以下 4 种操作:
- 1. 将磁头左移一位.
- 2. 将磁头右移一位.
- 3. 将磁头当前所在位置的值赋为 0 .
- 4. 将磁头当前所在位置的值赋为 1 .
- 现在给出一个长度为 len 的操作序列和某个状态，求有多少初始图灵机状态，满足存在一个磁头的初始位置，使得磁头在移动中始终不移出图灵机，且存在一个中间状态等于给出的状态?
- n <= 36, len <= 555

SRM555: Hard

- 考虑枚举磁头的初始位置，模拟进行操作，得到磁头是否会移出图灵机以及被修改的位置。如果被修改的位置不匹配，那么这种初始位置一定是不合法。

- 否则，考虑最后一次所有被修改的位置都能对应给出状态的时刻(此时被修改的位置一定最多)，所有被修改的位置，它们的初始状态可以“任选”，而未被修改的位置就要和给出状态一致。

- 假设有 x 个被修改的位置，此时的方案数为 2^x

SRM555: Hard

- 对于一种初始状态，可能有多个磁头初始位置能满足条件。
- 考虑容斥去重：
  - 枚举磁头初始位置的集合 S，得到若干个可以“任选”的位置集合
  - 取交集，设交集大小为 v。
  - 有 2^v 种初始状态，满足：任取 $i \in S$, 磁头从 i 开始进行操作，能在某一步转移到目标状态
  - 带上容斥系数 (-1)^(|S| + 1)，就能统计答案了
- 直接容斥复杂度为 O(2^n)，考虑优化

SRM555: Hard

- 我们记磁头活动区间的长度为 L，考虑：
  - 如果 L > n / 2，那么合法的磁头初始位置不超过 n / 2 个；
  - 如果 L < n / 2，那么枚举的磁头位置集合 S 中，如果有两个距离超过 L 的位置，“任选”位置集合一定为空；
- 综上，我们用搜索来进行容斥的过程，搜索的同时维护当前可以“任选”的位置集合，当集合为空时停止搜索，统计贡献，复杂度就很科学了。
- 总复杂度 O(n * 2^{n/2})

补集转化

- 一句话：先算出来所有的，然后减去不合法的

51nod 1486

- 一个 n * m 的棋盘，有 k 个坏格
- 每次向右或向下，从左上走到右下，不能经过坏格
- 方案数?
- n, m <= 10W, k <= 2k

51nod 1486

- 取出 k 个坏格，记 f[i] 表示从左上角**不经过**其它坏格，走到第 k 个坏格的方案数
- 考虑转移，走到 (x,y) 这个格，共有 C(x+y, x) 种方案
- 减掉不合法的，枚举第一次经过的坏格为 j
  - f[i] -= f[j] * C(x[i]+y[i]-x[j]-y[j], x[i] - x[j])
- 把终点也视为一个坏格，就能算方案数了

技巧

- 这种问题可以抽象为计算从初始状态转移到目标状态的方案数
- 给出一堆XXX规则，要求计算某些状态不能发生 / 一定发生 / 在YY首次发生的方案数
- f[i] 表示转移到 i 首次发生的方案数
- 可以算出转移到 i 的方案数，然后减去不合法的
- 对不合法的，枚举它首次发生在 j
    - f[i] -= f[j] * trans(j, i)

例2

- (n+1) * (m+1) 的棋盘，有k个格不能走
- A、B 两人从 (0, 0) 走到 (n, m)，每次向右或向下
- 路径不能有除了起点、终点之外的公共点
- 方案数?
- n, m <= 10W, k <= 2k

例2

- 记 a = (0, 1), b = (1, 0), x = (n-1, m), y = (n, m-1)
- 要求从 a->x, b->y， 且不相交的路径数
- 答案实际上等于 (a->x) * (b->y) - (a->y) * (b->x)
- 证明?
- 可以把每种不合法的方案与 (a->y) * (b->x) 中的某个方案对应起来

例3

- 考虑上一题的 General 的情况
- n 个点、m 条边的 DAG，计算从 a 到 b、c 到 d 的不相交路径数
- n <= 2k, m <= 4k

例3

- 有两种解法，时空复杂度均为 O(nm) - O(n^2)

例3

- 解法一
- 补集转化思想，枚举首次相遇减不合法方案
- 首先计算 G(i, j) 表示 i 到 j 的方案数
- 计算 F(v) 表示 a、c 走到 v，且不在 v 之前的点相遇的方案数
- ans = G(a,b) * G(c, d) - ∑ F(v) * G(v, b) * G(v, d)

例3

- 解法二
- 假设点标号就是拓扑序，不妨设 a > c
- 设 f[i][j] 表示 a、c 分别走到 i、j，且不相交的方案数
  - 规定 f[i][j] 能转移到 f[k][j] 当且仅当存在边 (i, k) 且 k > j
  - 规定 f[i][j] 能转移到 f[i][k] 当且仅当存在边 (j, k)，且 [k > i 或 i = a]
- 这个解法在没有模数的时候很优越

例4

- 从另一个角度考虑上一道题 General 的情况
- 现在有 n * n 的网格
- 指定了网格第一行的 k 个点，和最后一行的 k 个点
- 问用不相交的路径把这些点两两连起来的方案数
- （路径只能向右或向下）
- n <= 10W, k <= 300

例4

- Lindström–Gessel–Viennot lemma
- 造一个矩阵 M，M[i][j] 表示从上方第 i 个点到下方第 j 个点的方案数
- ans = det(M[i][j])
- 证明的大概思路是构造一种映射，使得每种不合法的方案一正一负相互抵消。具体可以去看维基

BZOJ 3622

- 给两组 n 个数 a[], b[]，保证数字互不相同，问有多少种将它们配对的方式，使得 a[i] > b[i] 的对数恰好为 k
- n, k <= 2000

BZOJ 3622

- 使用补集转化的思路
- 分两步递推：
  - 首先算出至少 k 组 a[i] > b[i] 的方案数
  - 递推算出恰好 k 组 a[i] > b[i] 的方案数

BZOJ 3622

- 首先两数组升序排序，记 num[i] 表示 a[i] 比 b[] 中多少元素大

- 设 f[i][j] 表示考虑 a[] 的前 i 个数中选出 j 个，再在 b[] 中选出 j 个使得对应 a[k] > b[k] 的方案数

- f[i][j] = f[i - 1][j] + f[i - 1][j - 1] * (num[j] - j + 1)

BZOJ 3622

- 设 g[i] 表示恰好 i 组 a[k] > b[k] 的方案数
- g[i] = f[n][i] * (n-i)! - ∑(i < j <= n) g[j] * C(j, i)
- 如果一时觉得难以理解的话，还是想像某个特定的、合法的或不合法的匹配会被上述式子计算多少次

PART 2: DP 技巧

- 这一部分的内容比较杂
- 涉及到一些OI中的计数、统计类问题的常用技巧
- 教大家更优雅地刻画状态、设计转移

平方处理

- 要计算的式子 / 贡献里有平方怎么办?
- 1. 可以考虑拆平方的括号
- 2. 假设有 n 个对象，n^2 的一种组合意义是：两个人从 n 个对象中独立地各选一个的方案数

51nod 1684

- 给出一种新的按位运算 ‘#’ 的真值表
- 定义一个序列的价值为序列中所有数从左向右进行 ‘#’ 运算得到的值
- 给定大小为 n 的序列 a[]，求它的 2^n-1 个非空子序列的价值平方和
- n <= 50000, a[i] < 2^30

51nod 1684

- 拆括号的技巧
- 二进制下，枚举每一位的平方，以及每两位的乘积对答案的贡献，使用一个简单DP统计即可。

BZOJ 1566

- 给两个0/1串 s1、s2，每次从某个串末尾取走一个字符，这些字符按顺序构造了一个新的0/1串，记为 s，对某个给定的串 s，若有 s_x 种取字符的方法能得到它，则它对答案有 {s_x}^2 的贡献，计算答案

- |s1|, |s2| <= 500

BZOJ 1566

- 相信大家都知道了怎么处理这个平方
- 想想两个人分别独立地取字符，有 {s_x}^2 种方案他们同时得到 s。
- 那么，要计算所有可能的 s_x 的平方和，可以转化为统计两个人取到的串相同的方案数
- 大力DP

分阶段转移

- 当你设计出来一个DP方程之后——
- 发现转移非常复杂?
- 此时，如果转移中可以发现明显的顺序性，那么可以考虑加维处理，在状态设计中存储更多信息，然后把原来的转移按照顺序拆成几个转移逐个考虑
- 可以以较小的代价明显地优化转移的复杂度

BZOJ 1138

- (并不是计数题)
- n 个点、m 条边的有向图，每条边上有一个字符
- 回答若干个询问，每次询问两点 x、y 之间的最短回文路长度
- n <= 400, m <= 6W

BZOJ 1138

- 设 f[x][y] 表示 x、y 之间的答案
- 转移可以卡成 O(n^2) 的样子…….
- 再设 g[x][y][c] 表示 x 到 y，x 这边多走了一个 c 的最短回文路
- 分两步转移，就没有上面提到的问题啦

BZOJ 3195

- 有 n 个点，你需要连 m 条边，使得每个点度数为偶数，同时给定参数 K，要求连边的两点 u, v 有 |u-v| <= K，允许重边，方案数?

- n, m <= 30, K <= 8

BZOJ 3195

- 这道题的套路还是比较容易看出来的
- 从前向后考虑每个点，设 f[i][j][S] 表示考虑到第 i 个点，连了 j 条边，前 K 个点的度数奇偶性为 S 的方案数
- 然后你发现...转移非常麻烦的样子?

BZOJ 3195

- 你可以给转移赋予一个顺序性
- f[i][j][S][l] 表示正在考虑 i 和 i-K+l 的连边
- 转移就可以优化到 O(1) 了

不记录无用的信息

- 我们DP的时候，需要让状态能准确刻画之前的决策对以后的所有影响
- 但是这个容易做过头
- 状态设计得太复杂，然后你发现根本没法转移
- 只好暴力分然后走人，GG
- 为了克服这一点，我们需要对题目条件有更深入的思考，充分挖掘可以利用的信息，不要记录无用的信息

51nod 1301

- 给定 n，m
- 问满足以下条件的集合对 A, B 的个数：
  - $A \subseteq \{1, 2 \dots n\}$, $B \subseteq \{1, 2 \dots m\}$
  - $A \cap B = \Phi$
  - A 中元素异或和 < B 中元素异或和
- n, m <= 2000

51nod 1301

- 我们可以非常自然地写一个 f[i][x][y] 表示分配了前 i 个数，两集合异或和分别为 x、y 的方案数
- 但显然这样做是 O(n^3) 的
- 怎么办呢?

51nod 1301

- A < B 意味着什么?
- 在二进制下，它们最高的若干位相同，然后到了某一位 A 是 0、B 是 1
- 最高若干位相同，而最低若干位不需要关心
- 也就是说这个状态设计是有很大优化空间的

51nod 1301

- O(log n) 枚举 A 和 B 最高的不同的二进制位 k
- f[x][0 / 1] 表示 A 和 B 异或和为 x，且 B 的第 k 位为 0/1 的方案数
- 刷表转移，然后统计答案即可
- 复杂度为 O(n^2\log n)，可以优化到 O(n^2)
- 值得一提的是，对“异或背包”的转移可以原地进行，不需要滚动数组，常数 (应该) 会更小

SRM 532 hard

- 用 K 种颜色给 n * m 的地图染色，使得任意两行都有至少一种颜色的数量不一样。求方案数
- n <= 10, K <= 50, m <= 100

SRM 532 hard

- 考虑题目的特点，从哪个角度入手比较方便?
    - 从小到大考虑每种颜色 x 的染色
    - 对颜色 x，依次考虑哪些行染了 1 个 x、2 个 x、3 个 x……依次类推
- 设 f[n][m][x][y] 表示现在有 n 行在颜色1 ~ x-1 的染色情况相同，每行有 m 个空可以用，正在考虑第 x 个颜色染 y 个的方案数

SRM 532 hard

- 转移枚举 c 行染了 y 个颜色 x：
  - f[n][m][x][y] +=
  - C(n, c) * C(m, y) ^ c *
  - f[c][m - y][x+1][0] * f[n - c][m][x][y + 1]

BZOJ 2169

- 考虑一张 n 个点的无向图，其中已经连了 m 条边，需要你再连 k 条边，使得所有点的度数为偶数；
- 连边要求：没有重边、自环，允许和已经存在的边重
- 求方案数
- n, m, k <= 1000

BZOJ 2169

- 每次连边改变两个点的奇偶性
- DP 过程中，我们最关心有多少点度数为奇数
- 设 f[i][j] 表示**无顺序**地连了 i 条边，有 j 个点的度数为奇数的方案数
- 不知道哪些位置已经连过边，能正确转移吗?

BZOJ 2169

转移：

$$
\begin{aligned}
f[i][j] = \frac{1}{i}(\, & f[i-1][j]*j*(n-j) \\
& + f[i-1][j+2]*C_{j+2}^{2} \\
& + f[i-1][j-2]*C_{n-j+2}^{2} \\
& - f[i-2][j]*(C_n^2-(i-2))\,)
\end{aligned}
$$

转移的前三项枚举三种不同的情况，最后减掉的那一项是去掉重复计数

请认真领会、透彻理解DP转移

BZOJ 4498

- 在长度为 L 的直线段的整点上安排站 n 个魔法师，每个魔法师有一个能量值 d[i]，每个魔法师的左右 d[i] 范围内不能有别的魔法师，求方案数

- L <= 10^6；n, d[i] <= 40

BZOJ 4498

- 首先考虑，魔法师的排列已经确定，有多少方案数?
- 设 w = ∑(1<=i<n) max(d[i], d[i+1])
- 则方案数可以通过组合数计算
- 稍有常识的同学就能看出，这个 w 不会太大
- 我们可以把每种魔法师排列的 w 统计出来，然后统一算答案

思考

- 对排列计数有两种姿势的样子...
  - 从前向后决定每个位置填什么
  - 从 小到大 / 大到小 决定每个数填哪里
  - 我们选择第二种
- 每个 d[i] 对 w 会有 0 / 1 / 2 倍的贡献，取决于排列中它两边数和它的大小关系
- 我们从大到小插入数字，插入每个数的时候就枚举它对 w 有多少贡献，并计算之

思考

- 把可以放数字的位置视为一个括号
- 初始时只有一个括号 (　)
- 考虑**从大到小**插入某个数 A 时，把它放在某个括号里：
  - 若它对答案有两份贡献，要求它两边都要有比它更小的数，把括号分裂成两个：( ) A ( )
  - 若它对答案有一份贡献，要求它某一边数比它小，把它贴着括号的边放：A ( ) or ( ) A
  - 若它对答案没有贡献，则它两边没有比它小的数，把括号删掉；

BZOJ 4498

- 顺着这个思路，设 f[i][j][k] 表示插入了 i 个数，目前 w 为 j，有 k 个括号的方案数
- 转移上边已经讲过了，可以 O(1)
- DP 完成以后组合数加一加就好了

PART 3: 图计数

- 这一部分关注一类重要的组合对象：图的计数
- 需要大家对递推和生成函数的相关知识有一定了解
- 关注图计数问题的切入点、巧妙地选择枚举对象以不重不漏地计数

图计数

- 图的计数问题几种 ~~技巧~~ 套路：
  - 用所有的对象减去不合法的对象
  - 取一个特殊点，枚举特殊点的状态
  - 根据图的特殊性找到其他一些方便枚举的东西
- ~~得到递推公式后，考虑能否 FFT 加速~~

无向连通图

- 给定 n，求 n 个点的无向连通图个数
- n <= 3W

无向连通图

- 设为 f[n]，再设 n 个点的图个数为 h[n]
- 递推，减掉不合法的，有公式：
- f[n] = h[n] - sum{i<n} C(n-1,i-1) * f[i] * h[n-i]
- 可以FFT加速的样子……

DAG 计数

- 给定 n，求 n 个点的 DAG 数目
- n <= 3W

DAG 计数

- 设为 f[n]
- 根据 DAG 的性质，我们枚举入度为 0 的点，有
- f[n] = ∑{k=1}^{n} C(n, k) * f[n - k] * 2^(k * (n - k))
- ……对吗?

DAG 计数

- 考虑一张 n 个点，有 k 个点入度为 0 的图
- 好像被计算了 2^k - 1 遍的样子…
- 改进一下，套一个容斥系数：
- f(n)= ∑ (-1)^(k-1)*C(n,k)*2^(k*(n-k))*f(n-k)

DAG 计数

- 这个也可以 FFT
- 把 2^{k * (n - k)} 中的 (k * (n - k))
- 拆为 1/2 * (n^2 - k^2 - (n - k) ^2)
- 就能分治FFT了
- 进一步推导还能多项式求逆

弱连通DAG计数

- 如题。
- n <= 1W，这是坠吼的

弱连通DAG计数

- 设全部为 f[n]，设连通为 g[n]
- g[n] = f[n] - ∑(k < n) g[k] * f[n - k] * C(n-1, k-1)
- 可以FFT加速的样子…
- 可以多项式求逆的样子…

强连通图

- 给定 n，求 n 个点的强连通图的个数

强连通图

- 设答案为 f[n]，设 h[n] 表示 n 个点的图数量
- 考虑用所有图减去不合法的，得到方案数
- 不合法的图，所有强连通分量缩掉之后可以得到一个有至少两个点的 DAG
- 我们枚举这个 DAG 里入度为 0 的点

强连通图

- 但是现在入度为 0 的“点”是一些连通分量缩来的…
- 我们可以设一个 g[n]，表示：
  - n 个点，组成 k 个强连通分量
  - 对 g[n] 的贡献为 (-1)^(k-1)
- g[n] = f[n] - ∑(k<n) C(n-1,k-1) * f[k] * g[n - k]

强连通图

- 现在可以推 f 了
- f[n] = h(n) - ∑(k < n) C(n,k)*2^(k*(n-k))*h(n-k)*g(k)
- 可以 FFT 的样子…
- 但是非常麻烦，能想出 O(n^2) 做法就能夸夸自己啦

点双连通图

- 我承认自己目前好像只会 O(n^3)
- 大家可以自己撕烤一下...

BZOJ 3812

- n 个点、m 条边的有向图
- 问有多少生成子图是强连通的
- n <= 15

BZOJ 3812

- 这道题涉及的所有 idea 在上面已经讲过了
- 不同点在于现在大小相同的点集不一定是等价的，把上题的状态设计改为点集，状压DP即可
- 复杂度可以而且必须做到 O(3^n)

PART 4：数学Trick

- 这一部分讲几个我自己感觉很妙的数学技巧
- 都是一些正在或者已经普及的科技
- 数学性较强，需要一些做题的经验

最值反演

- 我承认这个中二的名字是我乱取的
- 实质上是容斥的一种应用
- 对数组 a[]，有这样一个公式：

$$\max(a_i) = \sum_S (-1)^{|S|+1} \min(a_i \mid i \in S)$$

最值反演

- 两边的最值运算取反也成立
- 这个公式NB在哪儿呢?
  - 等式两边取期望也成立
  - 数论上，可以用 gcd 来反演 lcm
- 下面我们一起来看一下它能做什么事吧～

UOJ 214

- n 个小朋友站成一排，要教他们唱歌
- 有若干个课程 (x, y)，表示可以教小朋友 x 唱第 y 个音
- 给定一个长为 m 的乐曲，乐曲每位是一个音
- 每次上课，我们会等概率的选择一个课程 (x,y)，教学。注意一个课程可以被上多次
- 当 n 个小朋友中存在一个区间 [i, i+m)，满足对应位置的小朋友会唱乐曲对应位置的音，停止上课
- 问上课次数的期望
- m <= n <= 30，音用 26 个小写字符表示

UOJ 214

- 首先考虑，[1, m+1) 可以唱出来的期望?
- 这相当于给定 n 个球，有 k 个关键球，每次摸一个放回去，问 k 个关键球都被摸过的期望
- 答案是 $\sum_{i=1}^{k} \frac{n}{k-i+1}$

UOJ 214

- 这个方法可以扩展为“给定若干区间，要求这些区间都会唱的期望”

- 也即，可以实现取 E(max)

- 而题目要求 E(min)，我们可以 2^(n-m+1) 容斥，用 E(max) 来反演 E(min)

- 复杂度可以做到 O(2^(n-m+1) * n)

UOJ 214

- 当 m 小而 n 大的时候，上述方法不太管用的样子…
- m 小的时候可以用一种经典 DP 来辅助容斥！
- 写一个 f[i][S][k] 表示考虑了前 i 个位置，之前 m 个区间中必须满足的为 S，现在有 k 个教程必须上的方案数
- 个人认为这个DP的复杂度为 O(2^m * n^3)
- 折中一下，这道题的复杂度大概可以做到 O(2^{n/2} * n^2)

SRM 583：Hard

- 给出一个 n*m 的棋盘，每个格子有一个 0~9 的数字，记棋盘上数字和为 S，现在要染黑棋盘

- 每次会随机选一个格子染，写有 x 的格子有 x / S 的概率被选到，一个格子可以被染多次

- 问期望多少次之后，棋盘每行每列均有至少一个格子被染黑

- n, m <= 21, n * m <= 150

SRM 583：Hard

- 设随机变量：第 i 行至少一格染黑的时间为 X[i]，第 i 列至少一格被染黑的时间为 Y[i]
- 这道题要求期望的 E(max(X[i], Y[i]))
- 我们比较容易算的情况是 E(min)
- 用期望的 min 反演 max
- 行、列中较小的一维暴力容斥，另一维写成 DP

HDU 4624

- 现在有 n 个白球排成一行。
- 每次会均匀等概率地随机一个区间 [l, r]，把区间中的球染黑
- 问所有球都被染黑的期望
- n <= 50

HDU 4624

- 记 X[i] 为 “第 i 个球被染黑” 的时间期望，题目要求 E(max(X[i]))
- 对给定集合 S，可以计算 E(min(X[i] | i ∊ S))
    - 这实际上是 S 中的球标记了一些区间，问这些区间中至少一个被随到的期望步数
    - 设被标记了 num 个区间，那么期望为 C(n, 2) / num
- 尝试写一个DP，对所有可能的 S，统计 (-1)^|S| * min(X[i] | i ∊ S)
- f[i][k][0 / 1] 表示最后一个选择的球为 i，i 之前有 k 个可以选的区间，选的球数奇偶性为 0 / 1 的方案数
- （其实最后一维可以省掉）

51nod 1355

- 给定大小为 n 的数组 a[]
- 求 lcm(fib[a[1]], fib[a[1]]… fib[a[n]])
- 对 10^9 + 7 取模
- n <= 5W, a[i] <= 10^6

51nod 1355

- 首先有 gcd(fib[n], fib[m]) = fib[gcd(n, m)]
- 考虑每个质因子的贡献，可以得到集合的 gcd 反演 lcm 的公式
- 然后就可以想到 2^n 的容斥大法了
- …或许善良的出题人可以给 2^n 一点部分分吧…

51nod 1355

- 注意到 a[i] <= 10^6
- 这么小的范围内进行容斥，然后计算贡献，就可以考虑使用一些数论技巧
- 思路的核心在于计算一个函数 g，满足
  - $f[i] = \prod_{(d \mid i)} g[d]$
- 可以理解为“乘法版的莫比乌斯反演”，具体细节略

组合数？组合数！

- 组合数，又称二项式系数
- 广泛出现于各类OI计数问题中
- 有必要掌握一些把玩组合数的技巧

常用公式

$$\binom{n}{k}\binom{k}{i} = \binom{n}{i}\binom{n-i}{k-i}$$

$$\sum_{i=0}^{n}\binom{n}{i}a^i = (a+1)^n$$

$$\sum_{i=0}^{n}\binom{n}{i}a^i \cdot b^{n-i} = (a+b)^n$$

$$\sum_{i=0}^{k}(-1)^k \cdot \binom{n}{i} = (-1)^k \cdot \binom{n-1}{k}$$

常用公式

$$\binom{n}{k} + \binom{n}{k-1} = \binom{n+1}{k}$$

$$\binom{n}{m} \equiv \binom{n\%p}{m\%p} \cdot \binom{n/p}{m/p} \pmod{p}$$

$$\sum_{i=0}^{n} \binom{n}{i}^2 = \binom{2n}{n}$$

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

Card Game for three

- 字符集为 ‘a’, ‘b’, ‘c’
- 令 cnt[c][i] 表示串的前缀 str[1, i] 中有多少字符 c
- 有多少长度为 3^{n+m+k} 的串，满足：
- 存在一个 pos，使得
    - cnt[‘a’][pos] = n
    - cnt[‘b’][pos] <= m
    - cnt[‘c’][pos] <= k
- n, m, k <= 300000

Card Game for three

- 枚举这个 pos
- 用二项式系数算方案数
- 熟练应用杨辉(Pascal ?) 三角形

结论题

- 给定 n，求

$$\sum_{k=0}^{n} 2^k \binom{n}{k} \binom{n-k}{\lfloor \frac{n-k}{2} \rfloor}$$

推导

$$
\begin{aligned}
\text{原式} &= \sum_{k=0}^{n}\sum_{i=0}^{k}\binom{k}{i}\binom{n}{k}\binom{n-k}{(n-k)/2} \\
&= \sum_{i=0}^{n}\binom{n}{i}\sum_{k=i}^{n}\binom{n-i}{k-i}\binom{n-k}{(n-k)/2} \\
&= \sum_{i,a,b=0,1}\binom{n}{i}\binom{n-i}{a}\binom{n-i-a}{a+b}
\end{aligned}
$$

推导

$$
\begin{aligned}
&= \sum_{i,a} \binom{n}{i} \binom{n-i}{a} \binom{n-i-a+1}{a+1} \\
&= \sum_{h=0}^{n} \binom{n}{h} \sum_{a=0}^{h} \binom{n-h+1}{a+1} \binom{h}{h-a} \\
&= \sum_{h=0}^{n} \binom{n}{h} \binom{n+1}{h+1} \\
&= \binom{2n+1}{n}
\end{aligned}
$$

结论题

- 其实也可以用组合技巧证明，说明两边在对同一个对象计数即可。
- 当然这种题出到OI里会被裱的，不过这个推倒思路值得我们研究学习一下。

求和1

- 给定 n, k，求 $\sum (i \le n/k)\ C(n, ik)$
- $n \le 10^{18}, k \le 10^5$

求和1

- 从 n 个物件中取出恰好为 k 的倍数个物件的方案数
- 循环卷积，可以使用各种花式做法

求和2

- 给定 n, k，求 $\sum$ (i <= n/k) C(n-ik, i)
- n <= 10^18, k <= 10^5

求和2

- 从长度为 n 的区间中取出若干个、不相交的、长度均为 (k+1) 的区间的方案数
- ans[i] = ans[i - k - 1] + ans[i - 1]
- 常系数线性递推问题
- O(k^3\log n) -> O(k^2\log n) -> O(k\log k\log n)

组合数 & 多项式

- 众所周知，多项式有两种表达方式
- 实际上有第三种：
- f(x) = ∑(k <= n) a[k] * C(n, k)
- 这种表示方式和系数表示法的互相转化是唯一的

组合数 & 多项式

- 如果有 a[]，可以FFT把 f[] 整出来
- 如果有 f[]，可以二项式反演把 a[] 给整出来
- （二项式反演就是容斥那里的那个奇奇怪怪的式子

优雅地求和

- UOJ 269
- 给定 m 次多项式的 f[0] ~ f[m]，参数 n、k，求：
- ans = $\sum_{k=0}^{n} f(k)\binom{n}{k}x^k(1-x)^{n-k}$
- 模数 998244353

优雅地求和

首先可以二项式反演得到 $a[]$

$$
\begin{aligned}
ans &= \sum_{k=0}^{n} \binom{n}{k} \cdot x^k \cdot (1-x)^{n-k} \sum_{i=0}^{m} \binom{k}{i} \cdot a[i] \\
&= \sum_{i=0}^{m} a_i \sum_{k=i}^{n} \binom{n}{k} \binom{k}{i} \cdot x^k \cdot (1-x)^{n-k} \\
&= \sum_{i=0}^{m} \binom{n}{i} \cdot a_i \cdot x^i \sum_{k=i}^{n} \binom{n-i}{k-i} \cdot x^{k-i} \cdot (1-x)^{n-k} \\
&= \sum_{i=0}^{m} \binom{n}{i} \cdot a_i \cdot x^i
\end{aligned}
$$

插值

- 给定 n 次多项式 f(x) 的 f[0] ~ f[n]
- 求 f(k)?
- 求 ∑(i<=k) f[i] ?

一般情况够用的插值

- 定义序列的 0 阶差分 g0[x] = f[x]
- k 阶差分定义为 gk[x] = g(k-1)[x+1] - g(k-1)[x]
- 则 f[k] = ∑(i <= n) gi[0] * C(k, i)
- O(n^2)

模数感人时的插值

- FFT 把多项式的组合数表示的系数爆算出来
- O(n\log n)

高端插值

给定 $n+1$ 个点 $(i, f[i])$，记

$$p_j(x) = \frac{\prod_{i \neq j}(x-i)}{\prod_{i \neq j}(j-i)}$$

使用拉格朗日插值，可以得到

$$f(x) = \sum_{i=0}^{n} f(i) \cdot p_i(x)$$

高端插值

- 把 k 代入这个插值多项式即可
- 这个 p_i(x) 怎么算呢?
- 维护一些前缀积、后缀积即可
- ~~具体细节我忘了~~
- 大家自行脑补一下吧...
- 可以 O(n)

前缀和

- 有结论：k 次多项式的前缀和是一个 k+1 次多项式
- 所以前缀和也就能做了

这个有什么用?

- 一般来说，当题目给定了一个 n，然后让你对一个关于 n 的组合对象进行计数 / 对关于 n 的和式计算
- 然后我们通过大力观察、大胆猜想考后证明等方式，发现答案是一个关于 n 的 k 次多项式
- 如果多项式不太好求 / 你比较懒 / 不想数学推导的话，就可以用一种通用的方法解题——即，把 f[0] ~ f[k] 爆算出来，然后插值。

例

- 自然数 k 次幂和问题：
- 给定 n, k，求 ∑ (i <= n) i ^ k
- k <= 2K, n <= 1e9
- EXT: k <= 5W, mod = 998244353
- EXTEXT: k <= 20W

例

- （黑科技出现之前都是用Bernoulli多项式做的）
- 答案是一个关于 n 的 k+1 次多项式
- 把 0~k 的 i^k 求出来，求前缀和就可以得到插值用到的信息了
- i^k 是一个积性函数，使用欧拉筛，只需要对素数暴力计算，合数的答案可以拼出来
- 复杂度可以 O(n)
- （黑科技出现之后就非常无脑了）

PART 5：杂题

- 讲一些(我认为)还没有发展成为一种套路的题
- 在计数题里一些其他好用的工具 / 技巧:
  - 高斯消元与线性方程组相关
  - Stirling 数和 Bell 数
  - Catalan 数
  - 大力搜搜搜

51nod 1323

- 给定 n * n 的数字矩阵
- 你要从矩阵中选出一些数，满足
  - 每行选了奇数个
  - 每列选了奇数个
  - 所选的数的乘积是完全平方数
- 求方案数
- n <= 20

51nod 1323

- 对每个位置，有选或不选两种决策，可以视为一个 0-1 变量
- 三种限制实质上都是关于 n^2 个变量的线性方程
- 高斯消元得到自由元个数
- 然后就能统计方案数了

BZOJ 4671

- 定义两个结点数相同的图 G1 与图 G2 的异或为一个新的图 G
- 其中如果 (u, v) 在 G1 与 G2 中的出现次数之和为 1, 那么边 (u, v) 在 G 中, 否则这条边不在 G 中.
- 现在给定 s 个结点数为 n 的图 G[1…s]
- 设 S = {G1, G2, . . . , Gs}，请问 S 有多少个子集的异或为一个连通图?
- s = 60, n <= 10

BZOJ 4671

- O(bell(n)) 枚举节点**至少**被划分出多少连通块，消元算方案数
- 可以通过打表，计算出每种划分需要套上的容斥系数
- （实际上是Stirling反演式）
- 复杂度 O(bell(n) * 消元)

TheMagicMatrix

- 一个 n * n 的棋盘，已经有 k 个位置填了数
- 每个位置用一个 0 ～ 9 的数字填
- 要求每一行、每一列数字和的个位均相同
- 求方案数
- n <= 1000, k <= 10

TheMagicMatrix

- 如果有一行、一列没有填任何数，那么可以把其他位置全部填好，然后通过这一行一列调整，方案数可以直接计算
- 这样一来，只需要考虑 n <= 10 的情况
- 只有不超过 100 个变量，在模 2 和模 5 意义下分别解方程求自由元个数，由 CRT 合并答案即可

51nod 1327

- 有一个N行M列的棋盘，即该棋盘被分为N*M格。现在向棋盘中放棋子，每个格子中最多放一个棋子，也可以一个不放。放完棋子后需要满足如下要求：
  - 对于第i行来说，其从左往右的前left[i] 个格子（即最左侧的left[i] 个连续的格子）中恰好一共有1个棋子；
  - 对于第i行来说，其从右往左的前right[i]个格子（即最右侧的right[i]个连续的格子）中恰好一共有1个棋子；
  - 每一列至少有一个棋子
- 保证 left[i] + right[i] <= M
- 求方案数
- n <= 50, m <= 200

51nod 1327

- 一眼看过去，这道题可以入手的角度很多，但是状态的设计比较困难
- 从左到右逐列确定每一位填什么
- 把题目给出的左区间和右区间，分别按照右端点和左端点排序
- 设计一个DP：f[i][j][k] 表示填好前 i 列，有 j 个“左区间” 和 k 个 “右区间” 被填上的方案数
- 转移不太好描述清楚，我们来画一画……

51nod 1260

- 考虑这样的二叉树：
  - 每个内部节点同时有左儿子和右儿子
  - 恰好有 n 个叶子
- 对任意满足上述条件的二叉树，我们按照中序遍历把每个叶子分别标上 1~n，接下来，你可以任意交换每个内部节点的左右子树。之后，中序遍历这棵树，把每个叶子的标号按照访问到它的时间写下来形成一个 1~n 的排列。
- 求所有这样的二叉树经过任意交换后可以得到的不同的排列的个数。
- n <= 10^6

51nod 1260

- 记答案的生成函数为 F(x)
- 通过一番推导，可以得到一个类Catalan数的形式：
  - (F(x) ^ 2 + F(x)) * x + 1 = F(x)
  - F(x) = (1 - 2x - sqrt(x^2 - 6x - 1)) / 2x
- 把根号二项式展开一下，就能得到一个通项公式了

51nod 1556

- 有一个 1 * n 的数，固定第一个数为 1
- 其他位置填正整数，且每个数不能和它前一个数相差超过一
- 求方案数
- n <= 10^6

51nod 1556

- 标算是默慈金数
- 但是可以用 Catalan 数做

LEBOXES

- 有 n 个盒子，开启第 i 个盒子有 p[i] 的概率获得 v[i] 的钱，有 1-p[i] 的概率获取一颗钻石
- 有 m 个商品，第 i 个商品的售价是 C[i] 元 + D[i] 个钻石
- 有人开了所有盒子，然后拿着所有财产去买了**尽可能多**的商品
- 求他买到的商品的期望个数
- n, m <= 30, v[i], C[i] <= 10^9

LEBOXES

- 题目里要求的期望似乎根本没法拆开计算贡献...
- 考虑搜搜搜!

LEBOXES

- 我们先DP出一张表，chart[i][j] 表示拿着 i 个钻石、j 元钱最多能买多少商品
- ???.jpg
- 注意到钱这一维太大了，不要存钱
- chart[i][j] 表示拿着 i 个钻石，想买 j 个商品，至少要花多少钱
- 这张表比原来那张表更好用，而且完全存得下
- 跑背包即可

LEBOXS

- 现在来搜开宝箱的情况。
- 考虑 Meet-in-middle，设前半部分有 L 个商品
- 我们 2^L 搜索所有可能的情况，把得到 1~L 个钻石的所有可能情况及其概率，存在 L 个vector里，并且求前缀和
- 然后，我们 2^(n-L) 搜索后半部分箱子，结合前半部分的 vector 和 chart 这张表，统计贡献
- L 取比 15 大一点的数，就能过了

51nod 1446

- 有 n 个点，其中一些点是好的，每个好的点有一个价值
- 现在要把这 n 个点连成一棵树，定义一个点是非常好的，当且仅当它本身和它的一个邻居都是好的
- 对于一棵树，定义它的价值为所有非常好的点的价值和
- 求有多少价值<= maxValue 的树
- $n <= 40$, $maxValue <= 10^9$

51nod 1446

- f[i][j][k] 表示连好的树分别有 i,j,k 个非常好点、好点和一般点的方案数
- n 次高斯消元可以算出这张表
- 考虑枚举哪些点是非常好的点，满足它们的价值和 <= maxValue，查表计算贡献
- 2^40 代价地枚举会T，Meet-in-middle 优化之

谢谢大家

- 祝大家AK训练赛！
- Question are welcome
- QQ：745350128