组合计数

前置知识

组合数，卡特兰数，斯特林数，加法原理，乘法原理，容斥原理，经典组合模型。

ARC110D

给定一个长度为 n 的序列 a，请你对所有长度为 n 且总和不超过 m 的序列 b 求出 $\prod_{i=1}^{n} \binom{b_i}{a_i}$。

$n, a_i \leq 2000$，$m \leq 10^9$

ARC110D

考虑组合意义，相当于有 n 组，每组有 a_i 个黑球，要在每组中插入若干个白球，使得白球总数不超过 $m-\sum a_i$。

ARC110D

考虑组合意义，相当于有 n 组，每组有 a_i 个黑球，要在每组中插入若干个白球，使得白球总数不超过 $m-\sum a_i$。
把所有黑球排在一行，然后往其中插入 $n-1$ 个木板表示组与组之间的间隔，于是相当于把白球插入这些黑球与间隔之中。

ARC110D

考虑组合意义，相当于有 n 组，每组有 a_i 个黑球，要在每组中插入若干个白球，使得白球总数不超过 $m-\sum a_i$。
把所有黑球排在一行，然后往其中插入 $n-1$ 个木板表示组与组之间的间隔，于是相当于把白球插入这些黑球与间隔之中。
于是变为经典组合数问题，答案就是 $\binom{m+n}{\sum a_i+n}$。

ABC221H

给定 n, m，对于 $k = 1, 2, \ldots, n$，求出有多少大小为 k 的正整数集合，满足集合中元素核等于 n 且任意数出现次数不超过 m。

$n, m \leq 5000$，要求做到 $\mathrm{O}(n^2)$

ABC221H

如果没有 m 的限制，就是分拆数。

ABC221H

如果没有 m 的限制，就是分拆数。
我们把问题描述成一个二维图上的问题，那么就是二维图的横长为 k，纵长为 n，求有多少从 $(0,0)$ 走到 (k,n) 的向上向右路径满足不能连续向右走超过 m 次。

ABC221H

如果没有 m 的限制，就是分拆数。

我们把问题描述成一个二维图上的问题，那么就是二维图的横长为 k，纵长为 n，求有多少从 $(0,0)$ 走到 (k,n) 的向上向右路径满足不能连续向右走超过 m 次。

考虑把这个图反转 90 度，于是问题变成从 $(0,0)$ 走到 (n,k) 的向上向右路径满足不能连续向上走超过 m 次。

ABC221H

如果没有 m 的限制，就是分拆数。
我们把问题描述成一个二维图上的问题，那么就是二维图的横长为 k，纵长为 n，求有多少从 $(0,0)$ 走到 (k,n) 的向上向右路径满足不能连续向右走超过 m 次。
考虑把这个图反转 90 度，于是问题变成从 $(0,0)$ 走到 (n,k) 的向上向右路径满足不能连续向上走超过 m 次。
这时我们再进行 dp，设 $f_{i,j}$ 表示走到 (i,j) 的方案数，于是转移有
$f_{i,j} \to f_{i+1,k}(k \in [j, j+m])$，前缀和优化即可，时间复杂度 $\mathrm{O}(n^2)$。

CF1842G

有一个长度为 n 的序列 a，定义其权值为所有元素的乘积。
有 m 次操作，每次等概率选择一个 i 然后将序列 a 的 $[i,n]$ 这个后缀全部加 v。
求最后序列 a 权值的期望。

$n \le 5000$，$m, v, a_i \le 10^9$

CF1842G

构建一个 $n \times (m+1)$ 的表格 b，其中 $b_{0,i} = a_i$，第 j 行表示第 j 次操作加的后缀。于是权值就是 $\prod_{i=1}^{n}\left(\sum_{j=0}^{m} b_{j,i}\right)$

CF1842G

构建一个 $n \times (m+1)$ 的表格 b，其中 $b_{0,i} = a_i$，第 j 行表示第 j 次操作加的后缀。于是权值就是 $\prod_{i=1}^{n} \left(\sum_{j=0}^{m} b_{j,i} \right)$

根据组合意义，若干个括号内和的积表示每个括号内可以任选一个元素，然后乘起来的和。于是上面式子的意义就是对于 $i \in [1, n]$，对 i 选择一个 a_i 或第 j 次的 $b_{j,i}$，然后乘起来的和。

CF1842G

构建一个 $n\times(m+1)$ 的表格 b，其中 $b_{0,i}=a_i$，第 j 行表示第 j 次操作加的后缀。于是权值就是 $\prod_{i=1}^{n}\left(\sum_{j=0}^{m}b_{j,i}\right)$

根据组合意义，若干个括号内和的积表示每个括号内可以任选一个元素，然后乘起来的和。于是上面式子的意义就是对于 $i\in[1,n]$，对 i 选择一个 a_i 或第 j 次的 $b_{j,i}$，然后乘起来的和。

考虑 dp，设 $f_{i,j}$ 表示已经为 $[1,i]$ 做了选择，且目前被这些元素选择的行（操作）有 j 个。转移考虑第 $i+1$ 个元素是选择 a_{i+1} 还是 v，有：

CF1842G

构建一个 $n\times(m+1)$ 的表格 b，其中 $b_{0,i}=a_i$，第 j 行表示第 j 次操作加的后缀。于是权值就是 $\prod_{i=1}^{n}\left(\sum_{j=0}^{m} b_{j,i}\right)$

根据组合意义，若干个括号内和的积表示每个括号内可以任选一个元素，然后乘起来的和。于是上面式子的意义就是对于 $i\in[1,n]$，对 i 选择一个 a_i 或第 j 次的 $b_{j,i}$，然后乘起来的和。

考虑 dp，设 $f_{i,j}$ 表示已经为 $[1,i]$ 做了选择，且目前被这些元素选择的行（操作）有 j 个。转移考虑第 $i+1$ 个元素是选择 a_{i+1} 还是 v，有：

- 选择 a_{i+1}：$f_{i,j}\times a_{i+1}\to f_{i+1,j}$

CF1842G

构建一个 $n\times(m+1)$ 的表格 b，其中 $b_{0,i}=a_i$，第 j 行表示第 j 次操作加的后缀。于是权值就是 $\prod_{i=1}^{n}\left(\sum_{j=0}^{m}b_{j,i}\right)$

根据组合意义，若干个括号内和的积表示每个括号内可以任选一个元素，然后乘起来的和。于是上面式子的意义就是对于 $i\in[1,n]$，对 i 选择一个 a_i 或第 j 次的 $b_{j,i}$，然后乘起来的和。

考虑 dp，设 $f_{i,j}$ 表示已经为 $[1,i]$ 做了选择，且目前被这些元素选择的行（操作）有 j 个。转移考虑第 $i+1$ 个元素是选择 a_{i+1} 还是 v，有：

- 选择 a_{i+1}：$f_{i,j}\times a_{i+1}\to f_{i+1,j}$
- 选择 v，但是选之前选过的行：$f_{i,j}\times j\to f_{i+1,j}$

CF1842G

构建一个 $n \times (m+1)$ 的表格 b，其中 $b_{0,i} = a_i$，第 j 行表示第 j 次操作加的后缀。于是权值就是 $\prod_{i=1}^{n}\left(\sum_{j=0}^{m} b_{j,i}\right)$

根据组合意义，若干个括号内和的积表示每个括号内可以任选一个元素，然后乘起来的和。于是上面式子的意义就是对于 $i \in [1,n]$，对 i 选择一个 a_i 或第 j 次的 $b_{j,i}$，然后乘起来的和。

考虑 dp，设 $f_{i,j}$ 表示已经为 $[1,i]$ 做了选择，且目前被这些元素选择的行（操作）有 j 个。转移考虑第 $i+1$ 个元素是选择 a_{i+1} 还是 v，有：

- 选择 a_{i+1}：$f_{i,j} \times a_{i+1} \to f_{i+1,j}$
- 选择 v，但是选之前选过的行：$f_{i,j} \times j \to f_{i+1,j}$
- 选择 v，但是选之前没选过的行：$f_{i,j} \times (m-j) \times \frac{i}{n} \to f_{i+1,j+1}$

AGC001E

有 n 个数对 $(A_i; B_i)$，求出

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\binom{a_i+b_i+a_j+b_j}{a_i+a_j}$$

$n \le 2\times 10^5$，$1 \le a_i, b_i \le 2000$

AGC001E

考虑 $\binom{a_i+b_i+a_j+b_j}{a_i+a_j}$ 的组合意义，即在一个二维图上从 $(0,0)$ 开始，只能向上向右走，走到 (a_i+a_j,b_i+b_j) 的方案数。

AGC001E

考虑 $\binom{a_i+b_i+a_j+b_j}{a_i+a_j}$ 的组合意义，即在一个二维图上从 $(0,0)$ 开始，只能向上向右走，走到 (a_i+a_j, b_i+b_j) 的方案数。
把这条路径整体平移，可以变成 $(-a_i,-b_i)$ 走到 (a_j,b_j) 的方案数，这样起点终点就分别独立了。

AGC001E

考虑 $\binom{a_i+b_i+a_j+b_j}{a_i+a_j}$ 的组合意义，即在一个二维图上从 $(0,0)$ 开始，只能向上向右走，走到 (a_i+a_j, b_i+b_j) 的方案数。
把这条路径整体平移，可以变成 $(-a_i,-b_i)$ 走到 (a_j,b_j) 的方案数，这样起点终点就分别独立了。
注意到值域非常小！于是我们可以直接 dp，设 $f_{i,j}$ 表示从位于 (i,j) 左下角的起点开始走到 (i,j) 的方案数，直接暴力转移，在可能为终点的地方统计答案即可。

AGC001E

考虑 $\binom{a_i+b_i+a_j+b_j}{a_i+a_j}$ 的组合意义，即在一个二维图上从 $(0,0)$ 开始，只能向上向右走，走到 (a_i+a_j, b_i+b_j) 的方案数。
把这条路径整体平移，可以变成 $(-a_i,-b_i)$ 走到 (a_j,b_j) 的方案数，这样起点终点就分别独立了。
注意到值域非常小！于是我们可以直接 dp，设 $f_{i,j}$ 表示从位于 (i,j) 左下角的起点开始走到 (i,j) 的方案数，直接暴力转移，在可能为终点的地方统计答案即可。
这样可能会多算进自己对自己的贡献以及其他贡献算入两边，特殊处理一下即可。

CF1204E

求有 n 个 1 和 m 个 -1 构成的序列的最大前缀和的和。

$n, m \leq 10^6$

CF1204E

令 f_i 表示最大值至少为 i 的方案数，那么答案就是 $\sum f_i$。

CF1204E

令 f_i 表示最大值至少为 i 的方案数，那么答案就是 $\sum f_i$。
求 f_i，考虑画一个二维图，令点 (i, sm_i) 表示前缀 i 和为 sm_i 的状态，那么一个序列对应一个起点为 $(0,0)$，终点为 $(n+m, n-m)$ 的路径，只能从 (x,y) 走到 $(x+1, y+1)$ 或 $(x+1, y-1)$。

CF1204E

令 f_i 表示最大值至少为 i 的方案数，那么答案就是 $\sum f_i$。
求 f_i，考虑画一个二维图，令点 (i, sm_i) 表示前缀 i 和为 sm_i 的状态，那么一个序列对应一个起点为 $(0,0)$，终点为 $(n+m, n-m)$ 的路径，只能从 (x,y) 走到 $(x+1, y+1)$ 或 $(x+1, y-1)$。
为了保证最大值至少为 i，那么就是要求这条折线经过直线 $y=x$。

CF1204E

令 f_i 表示最大值至少为 i 的方案数，那么答案就是 $\sum f_i$。
求 f_i，考虑画一个二维图，令点 (i, sm_i) 表示前缀 i 和为 sm_i 的状态，那么一个序列对应一个起点为 $(0,0)$，终点为 $(n+m, n-m)$ 的路径，只能从 (x,y) 走到 $(x+1, y+1)$ 或 $(x+1, y-1)$。
为了保证最大值至少为 i，那么就是要求这条折线经过直线 $y=x$。
首先如果起点终点在 $y=x$ 的两侧，那么一定会经过；否则如果在同侧，那么考虑把终点做 $y=x$ 的对称点，这样得到的路径一定经过，但终点不一定合法，但是如果我们找到第一个经过点，将后面的路径做 $y=x$ 的对称，终点就对了。换句话说，这两者构成双射。

CF1204E

令 f_i 表示最大值至少为 i 的方案数，那么答案就是 $\sum f_i$。
求 f_i，考虑画一个二维图，令点 (i, sm_i) 表示前缀 i 和为 sm_i 的状态，那么一个序列对应一个起点为 $(0,0)$，终点为 $(n+m, n-m)$ 的路径，只能从 (x, y) 走到 $(x+1, y+1)$ 或 $(x+1, y-1)$。
为了保证最大值至少为 i，那么就是要求这条折线经过直线 $y=x$。
首先如果起点终点在 $y=x$ 的两侧，那么一定会经过；否则如果在同侧，那么考虑把终点做 $y=x$ 的对称点，这样得到的路径一定经过，但终点不一定合法，但是如果我们找到第一个经过点，将后面的路径做 $y=x$ 的对称，终点就对了。换句话说，这两者构成双射。
两种情况都是一个组合数即可解决。

总结与练习题

总结与练习题

例题其实没有找到很多，但是这类题 Atcoder 上印象里挺常见的。

[NOI Online #2 提高组] 游戏

有一个 $n=2m$ 个点的树，其中 m 个黑点 m 个白点，你需要给这些黑点和白点分别标上 $1\sim m$ 的号。
请你对每个 $k=1,2,\ldots,m$，求出有多少标号方案满足恰好有 k 个 i，满足标号为 i 的黑点和白点互为祖孙关系。

$m\le 2500$

[NOI Online #2 提高组] 游戏

发现恰好是困难的，但是钦定是简单的，于是设 $F(i)$ 表示恰好 i 个关系的方案数，$G(i)$ 表示钦定 i 个关系的方案数，根据二项式反演有

$$F(i)=\sum_{j=k}^{n}(-1)^{j-i}\binom{j}{i}G(j)$$

[NOI Online #2 提高组] 游戏

发现恰好是困难的，但是钦定是简单的，于是设 $F(i)$ 表示恰好 i 个关系的方案数，$G(i)$ 表示钦定 i 个关系的方案数，根据二项式反演有

$$F(i) = \sum_{j=k}^{n} (-1)^{j-i} \binom{j}{i} G(j)$$

问题变成求 $G(i)$，考虑 dp，设 $f_{i,j}$ 表示以 i 为根时的 $G(j)$，转移如果不考虑根就是树上背包，考虑的话就讨论根是否与子树中的某个异色点匹配加入钦定的集合，注意到子树内异色点的数量只与 j 有关，故可以直接转移。时间复杂度 $\mathrm{O}(n^2)$。

AGC058D

求有多少长度为 $a+b+c$ 的只包含 ABC 三个字母的字符，满足：

- 恰好有 a 个 A，b 个 B，c 个 C。
- 不包含形如 ABC,BCA,CAB 的连续子串。

$a, b, c \leq 10^6$

AGC058D

不包含是困难的，但是包含看上去相对容易些，所以考虑容斥，钦定出现非法子串的位置。

AGC058D

不包含是困难的，但是包含看上去相对容易些，所以考虑容斥，钦定出现非法子串的位置。

看上去还是不好做，不过如果我们令 A, B, C 分别是 $0, 1, 2$，那么非法子串就是长度大于 3 的满足 $s_i + 1 \equiv s_{i+1} \pmod 3$，因此非法连续段具有传递性，我们可以转为钦定非法连续段。

AGC058D

不包含是困难的，但是包含看上去相对容易些，所以考虑容斥，钦定出现非法子串的位置。
看上去还是不好做，不过如果我们令 A, B, C 分别是 $0, 1, 2$，那么非法子串就是长度大于 3 的满足 $s_i + 1 \equiv s_{i+1} \pmod 3$，因此非法连续段具有传递性，我们可以转为钦定非法连续段。
但是钦定非法连续段看上去还是很困难，于是我们再进一步，转为钦定非法连续段的起点。

AGC058D

不包含是困难的，但是包含看上去相对容易些，所以考虑容斥，钦定出现非法子串的位置。
看上去还是不好做，不过如果我们令 A, B, C 分别是 $0, 1, 2$，那么非法子串就是长度大于 3 的满足 $s_i + 1 \equiv s_{i+1} \pmod 3$，因此非法连续段具有传递性，我们可以转为钦定非法连续段。
但是钦定非法连续段看上去还是很困难，于是我们再进一步，转为钦定非法连续段的起点。
如果一个位置 p 是一个非法连续段的起点，那么要满足：

- $s_{p-1} + 1 \not\equiv s_p \pmod 3$
- $\forall p \le i \le p+2, s_i + 1 \equiv s_{i+1} \pmod 3$

AGC058D

不包含是困难的，但是包含看上去相对容易些，所以考虑容斥，钦定出现非法子串的位置。

看上去还是不好做，不过如果我们令 A, B, C 分别是 $0, 1, 2$，那么非法子串就是长度大于 3 的满足 $s_i + 1 \equiv s_{i+1} \pmod 3$，因此非法连续段具有传递性，我们可以转为钦定非法连续段。

但是钦定非法连续段看上去还是很困难，于是我们再进一步，转为钦定非法连续段的起点。

如果一个位置 p 是一个非法连续段的起点，那么要满足：

- $s_{p-1} + 1 \not\equiv s_p \pmod 3$
- $\forall p \le i \le p+2, s_i + 1 \equiv s_{i+1} \pmod 3$

考虑后者要求我们把 ABC 三个字符捆绑起来，而如果我们先确定 s_{p-1}，ABC 三个字符的三个顺序中恰好有两个是合法的。

AGC058D

不包含是困难的，但是包含看上去相对容易些，所以考虑容斥，钦定出现非法子串的位置。
看上去还是不好做，不过如果我们令 A, B, C 分别是 $0, 1, 2$，那么非法子串就是长度大于 3 的满足 $s_i + 1 \equiv s_{i+1} \pmod 3$，因此非法连续段具有传递性，我们可以转为钦定非法连续段。
但是钦定非法连续段看上去还是很困难，于是我们再进一步，转为钦定非法连续段的起点。
如果一个位置 p 是一个非法连续段的起点，那么要满足：

- $s_{p-1} + 1 \not\equiv s_p \pmod 3$
- $\forall p \le i \le p + 2, s_i + 1 \equiv s_{i+1} \pmod 3$

考虑后者要求我们把 ABC 三个字符捆绑起来，而如果我们先确定 s_{p-1}，ABC 三个字符的三个顺序中恰好有两个是合法的。
于是我们以这样的方法计算：先把没有被捆绑的字符随意排列（这是一个组合数），然后选择若干个间隙插入捆绑的 ABC（这也是组合数），而插入间隙的前一个字符决定了插入 ABC 的顺序有恰好两种选择（这是一个 $2^{\text{起点数}}$）。

AGC058D

不包含是困难的，但是包含看上去相对容易些，所以考虑容斥，钦定出现非法子串的位置。
看上去还是不好做，不过如果我们令 A, B, C 分别是 $0, 1, 2$，那么非法子串就是长度大于 3 的满足 $s_i + 1 \equiv s_{i+1} \pmod 3$，因此非法连续段具有传递性，我们可以转为钦定非法连续段。
但是钦定非法连续段看上去还是很困难，于是我们再进一步，转为钦定非法连续段的起点。
如果一个位置 p 是一个非法连续段的起点，那么要满足：

- $s_{p-1} + 1 \not\equiv s_p \pmod 3$
- $\forall p \le i \le p+2, s_i + 1 \equiv s_{i+1} \pmod 3$

考虑后者要求我们把 ABC 三个字符捆绑起来，而如果我们先确定 s_{p-1}，ABC 三个字符的三个顺序中恰好有两个是合法的。
于是我们以这样的方法计算：先把没有被捆绑的字符随意排列（这是一个组合数），然后选择若干个间隙插入捆绑的 ABC（这也是组合数），而插入间隙的前一个字符决定了插入 ABC 的顺序有恰好两种选择（这是一个 $2^{\text{起点数}}$）。
注意特殊处理钦定起点位于序列开头的情况。

经典题

给定一个 n 个点的图和若干条边，求有多少边定向方案使得图成为一个 DAG。

$n \leq 18$

经典题

考虑 dp，设 f_S 表示 S 这个集合中的点的导出子图定向为 DAG 的方案数，转移可以考虑剥掉当前 DAG 的 0 度点。

经典题

考虑 dp，设 f_S 表示 S 这个集合中的点的导出子图定向为 DAG 的方案数，转移可以考虑剥掉当前 DAG 的 0 度点。
令当前点集 S 的 0 度点集合 T，那么要满足的是 T 内部没有边，$T \to S - T$ 随意，$S - T \to T$ 不能有边，S 内部是一个 DAG。

经典题

考虑 dp，设 f_S 表示 S 这个集合中的点的导出子图定向为 DAG 的方案数，转移可以考虑剥掉当前 DAG 的 0 度点。
令当前点集 S 的 0 度点集合 T，那么要满足的是 T 内部没有边，$T \rightarrow S-T$ 随意，$S-T \rightarrow T$ 不能有边，S 内部是一个 DAG。
可是上述条件只能保证 T 是 0 度，不能保证 $S-T$ 没有 0 度，于是这就变成了钦定，考虑容斥式子

经典题

考虑 dp，设 f_S 表示 S 这个集合中的点的导出子图定向为 DAG 的方案数，转移可以考虑剥掉当前 DAG 的 0 度点。

令当前点集 S 的 0 度点集合 T，那么要满足的是 T 内部没有边，$T \to S - T$ 随意，$S - T \to T$ 不能有边，S 内部是一个 DAG。

可是上述条件只能保证 T 是 0 度，不能保证 $S - T$ 没有 0 度，于是这就变成了钦定，考虑容斥式子

$$f_S = \sum_{T \subseteq S, T \neq \varnothing} {}_{P \in T} (-1)^{|T|-|P|} \cdot 2^{ways(P, S-P)} \cdot f_P$$
$$= _{P \in S} (-1)^{|P|-1} \cdot 2^{ways(P, S-P)} \cdot f_P$$

经典题

考虑 dp，设 f_S 表示 S 这个集合中的点的导出子图定向为 DAG 的方案数，转移可以考虑剥掉当前 DAG 的 0 度点。
令当前点集 S 的 0 度点集合 T，那么要满足的是 T 内部没有边，$T \to S-T$ 随意，$S-T \to T$ 不能有边，S 内部是一个 DAG。
可是上述条件只能保证 T 是 0 度，不能保证 $S-T$ 没有 0 度，于是这就变成了钦定，考虑容斥式子

$$f_S = \sum_{T \subseteq S, T \neq \varnothing} {}_{P \in T}(-1)^{|T|-|P|} \cdot 2^{ways(P,S-P)} \cdot f_P$$

$$= {}_{P \in S} (-1)^{|P|-1} \cdot 2^{ways(P,S-P)} \cdot f_P$$

时间复杂度 $O(3^n)$

P4707 重返现世

有 n 种原料，每个单位时间有 $\frac{p_i}{m}$ 的概率获得第 i 种原料，求得到 k 种不同原料的期望时间。

$n \le 1000, |n-k| \le 10, m \le 10000$。

P4707 重返现世

等价于求第 $n-k+1$ 大的出现时刻，可以使用扩展 min-max 容斥：

$$E(\max_k(S)) = \sum_{T\in S}\binom{|T|-1}{k-1}(-1)^{|T|-k}E(\min(T))$$

于是转化为求 min 的期望，而一个集合的 min 也就是求这个集合种元素第一次被选中的概率，也就是 $\frac{\sum_{x\in T} p_x}{m}$。

P4707 重返现世

等价于求第 $n-k+1$ 大的出现时刻，可以使用扩展 min-max 容斥：

$$E(\max_k(S)) = \sum_{T\in S}\binom{|T|-1}{k-1}(-1)^{|T|-k}E(\min(T))$$

于是转化为求 min 的期望，而一个集合的 min 也就是求这个集合种元素第一次被选中的概率，也就是 $\frac{\sum_{x\in T} p_x}{m}$。
设 $dp_{i,j,k}$ 表示考虑了前 i 个，$\sum_{x\in T} p_x$ 为 j，时

$$\sum_{T\in S}\binom{|T|-1}{k-1}(-1)^{|T|-k}$$

的值。

P4707 重返现世

从前 $i-1$ 个元素转移到 i 可以考虑第 i 个是否加入，如果不加入，转移就是 $dp_{i,j,k}=dp_{i-1,j,k}$。

P4707 重返现世

从前 $i-1$ 个元素转移到 i 可以考虑第 i 个是否加入，如果不加入，转移就是 $dp_{i,j,k} = dp_{i-1,j,k}$。
考虑如果加入第 i 个元素，那么考虑下面这个式子：

$$\binom{|T|-1}{k-1}(-1)^{|T|-k} = -\binom{|T|-2}{k-1}(-1)^{(|T|-1)-k} + \binom{|T|-2}{k-2}(-1)^{(|T|-1)-(k-1)}$$

放在转移式上就是 $dp_{i,j,k} = dp_{i-1,j-p_i,k-1} - dp_{i-1,j-p_i,k}$。

P4707 重返现世

从前 $i-1$ 个元素转移到 i 可以考虑第 i 个是否加入，如果不加入，转移就是 $dp_{i,j,k}=dp_{i-1,j,k}$。
考虑如果加入第 i 个元素，那么考虑下面这个式子：

$$\binom{|T|-1}{k-1}(-1)^{|T|-k}=-\binom{|T|-2}{k-1}(-1)^{(|T|-1)-k}+\binom{|T|-2}{k-2}(-1)^{(|T|-1)-(k-1)}$$

放在转移式上就是 $dp_{i,j,k}=dp_{i-1,j-p_i,k-1}-dp_{i-1,j-p_i,k}$。
合起来是 $dp_{i,j,k}=dp_{i-1,j,k}+dp_{i-1,j-p_i,k-1}-dp_{i-1,j-p_i,k}$。
加上滚动数组和一些细节即可通过。

[JLOI2015] 骗我呢

求有一个 $n \times m$ 的网格图 $a_{n,m}$ 满足：

- $a_{i,j-1} < a_{i,j}$
- $a_{i-1,j-1} < a_{i,j}$
- $a_{i,j} \in [0, m]$

$n, m \le 10^6$

[JLOI2015] 骗我呢

注意值域，相当于每行只有一个位置相差 2，其余都相差 1。

[JLOI2015] 骗我呢

注意值域，相当于每行只有一个位置相差 2，其余都相差 1。
令 p_i 表示第 i 行相差 2 的位置，那么要满足 $p_i + 1 \geq p_{i+1}$。

[JLOI2015] 骗我呢

注意值域，相当于每行只有一个位置相差 2，其余都相差 1。
令 p_i 表示第 i 行相差 2 的位置，那么要满足 $p_i + 1 \geq p_{i+1}$。
再转化一下，把第 i 行右移 $m - i$ 格，那么就是满足 $p_i \geq p_{i+1}$，以 $(n, 0)$ 为原点建立坐标系，那么一盒合法方案就是一个从 $(0, 0)$ 走到 $(n + m + 1, n)$ 且不经过 $y = x + 1$ 和 $y = x - (m + 2)$ 这两条线。

[JLOI2015] 骗我呢

注意值域，相当于每行只有一个位置相差 2，其余都相差 1。
令 p_i 表示第 i 行相差 2 的位置，那么要满足 $p_i+1 \geq p_{i+1}$。
再转化一下，把第 i 行右移 $m-i$ 格，那么就是满足 $p_i \geq p_{i+1}$，以 $(n,0)$ 为原点建立坐标系，那么一盒合法方案就是一个从 $(0,0)$ 走到 $(n+m+1,n)$ 且不经过 $y=x+1$ 和 $y=x-(m+2)$ 这两条线。
考虑容斥，用总方案数 − 首次经过的直线是上直线 − 首次经过的直线是下直线，两者类似，只讨论前者。

[JLOI2015] 骗我呢

注意值域，相当于每行只有一个位置相差 2，其余都相差 1。
令 p_i 表示第 i 行相差 2 的位置，那么要满足 $p_i + 1 \geq p_{i+1}$。
再转化一下，把第 i 行右移 $m - i$ 格，那么就是满足 $p_i \geq p_{i+1}$，以 $(n, 0)$ 为原点建立坐标系，那么一盒合法方案就是一个从 $(0, 0)$ 走到 $(n + m + 1, n)$ 且不经过 $y = x + 1$ 和 $y = x - (m + 2)$ 这两条线。
考虑容斥，用总方案数 − 首次经过的直线是上直线 − 首次经过的直线是下直线，两者类似，只讨论前者。
考虑使用前面对称的 trick，但是这样有问题，可能算入首次经过的直线是下直线的情况，于是再容斥，减去经过顺序为下上的直线，即把终点两次对称。
而这个可能仍不合法，于是再加上上下上的直线，以此类推。

[JLOI2015] 骗我呢

注意值域，相当于每行只有一个位置相差 2，其余都相差 1。
令 p_i 表示第 i 行相差 2 的位置，那么要满足 $p_i+1 \ge p_{i+1}$。
再转化一下，把第 i 行右移 $m-i$ 格，那么就是满足 $p_i \ge p_{i+1}$，以 $(n,0)$ 为原点建立坐标系，那么一盒合法方案就是一个从 $(0,0)$ 走到 $(n+m+1,n)$ 且不经过 $y=x+1$ 和 $y=x-(m+2)$ 这两条线。
考虑容斥，用总方案数 − 首次经过的直线是上直线 − 首次经过的直线是下直线，两者类似，只讨论前者。
考虑使用前面对称的 trick，但是这样有问题，可能算入首次经过的直线是下直线的情况，于是再容斥，减去经过顺序为下上的直线，即把终点两次对称。
而这个可能仍不合法，于是再加上上下上的直线，以此类推。
注意到对称 $O(n+m)$ 次后必定对称到二四象限（即无解），所以时间复杂度为 $O(n+m)$。

[UNR #2] 梦中的题面（加强版）

给定 n, m, b, c，求序列 $x_1, x_2, \ldots, x_m$ 的数量，满足：

- $\forall i, 0 \le x_i \le b^i - c$
- $\sum x_i \le n$

$m \le 300$，$3 \le b \le 10^9$，$-b+2 \le c \le b-1$，$1 \le n \le b^{m+1}$

[UNR #2] 梦中的题面（加强版）

第二个条件是好满足的，一个组合数即可计算。但是第一个不好满足，于是容斥。

[UNR #2] 梦中的题面（加强版）

第二个条件是好满足的，一个组合数即可计算。但是第一个不好满足，于是容斥。

令 $a_i = b^i - c + 1$，枚举集合 S 表示钦定不满足第一个条件的位置集合，那么有

$$ans = \sum_{S \subseteq \{1,2,\dots,m\}} (-1)^{|S|} \binom{n - \sum_{i \in S} a_i + m}{m}$$

[UNR #2] 梦中的题面（加强版）

第二个条件是好满足的，一个组合数即可计算。但是第一个不好满足，于是容斥。

令 $a_i = b^i - c + 1$，枚举集合 S 表示钦定不满足第一个条件的位置集合，那么有

$$ans = \sum_{S \subseteq \{1,2,\ldots,m\}} (-1)^{|S|} \binom{n - \sum_{i \in S} a_i + m}{m}$$

直接枚举 S 肯定是不行的，那我们尝试 dp？那后面的那坨组合数咋办呢。

[UNR #2] 梦中的题面（加强版）

第二个条件是好满足的，一个组合数即可计算。但是第一个不好满足，于是容斥。

令 $a_i = b^i - c + 1$，枚举集合 S 表示钦定不满足第一个条件的位置集合，那么有

$$ans = \sum_{S \subseteq \{1,2,\ldots,m\}} (-1)^{|S|} \binom{n - \sum_{i \in S} a_i + m}{m}$$

直接枚举 S 肯定是不行的，那我们尝试 dp？那后面的那坨组合数咋办呢。

注意到范德蒙恒等式

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

于是可以令 $f_{i,j}$ 表示考虑了前 i 个元素，组合数下指标是 j 的方案数，可以直接转移。（注意这里的组合数要推广到实数域上）

[UNR #2] 梦中的题面（加强版）

但是如果我们将组合数推广到负数域，问题又来了，如果 $n-\sum_{i\in S} a_i + m < 0$ 那么组合数本应为 0，但是会算成实数域上的组合数。

[UNR #2] 梦中的题面（加强版）

但是如果我们将组合数推广到负数域，问题又来了，如果 $n-\sum_{i\in S}a_i+m<0$ 那么组合数本应为 0，但是会算成实数域上的组合数。注意到 $b\geq 3$，那么有 $a_i>\sum_{j<i}a_j$，于是我们将 $n+m$ 化为 a 进制（即从 a_m 开始能减就减），那么满足要求就是 $\sum_{i\in S}a_i$ 化为 a 进制后不超过 $n+m$ 的 a 进制。

[UNR #2] 梦中的题面（加强版）

但是如果我们将组合数推广到负数域，问题又来了，如果 $n-\sum_{i\in S}a_i+m<0$ 那么组合数本应为 0，但是会算成实数域上的组合数。
注意到 $b\geq 3$，那么有 $a_i>\sum_{j<i}a_j$，于是我们将 $n+m$ 化为 a 进制（即从 a_m 开始能减就减），那么满足要求就是 $\sum_{i\in S}a_i$ 化为 a 进制后不超过 $n+m$ 的 a 进制。
直接在原 dp 的基础上套用一个数位 dp 即可。时间复杂度 $\mathrm{O}(m^3)$，瓶颈在于转移，可以用 FFT 优化至 $\mathrm{O}(n^2\log n)$

总结与练习题

总结与练习题

- CF997C Sky Full of Stars
- AGC041F Histogram Rooks
- P4921 烧情侣
- HAOI2015 按位或
- PKUWC2018 随机游走
- 清华集训 2014 主旋律
- AGC060D Same Descent Set

CF961G

有 n 个物品，每个物品有一个权值 w_i，定义一个集合 S 的权值为 $|S|\sum_{x\in S} w_x$，一个划分的权值为所有划分出来的集合的权值之和。求 n 个物品划分为 k 个集合的所有方案的权值和。

$n,k \leq 2\times 10^5$

CF961G

首先可以直接写出答案的式子

$$\sum_{i=1}^{n} w_i \sum_{j=1}^{n} j \cdot \binom{n-1}{j-1} \left\{ {n-j \atop k-1} \right\}$$

关键在于求后面那一坨。

CF961G

首先可以直接写出答案的式子

$$\sum_{i=1}^{n} w_i \sum_{j=1}^{n} j \cdot \binom{n-1}{j-1} \left\{ {n-j \atop k-1} \right\}$$

关键在于求后面那一坨。
考虑组合意义，即把 n 分为 k 个集合，包含元素 1 的集合大小之和。由于所有元素等价，于是可以转为算包含每个点的大小之和，最后除 n。

CF961G

首先可以直接写出答案的式子

$$\sum_{i=1}^{n} w_i \sum_{j=1}^{n} j \cdot \binom{n-1}{j-1} \left\{ {n-j \atop k-1} \right\}$$

关键在于求后面那一坨。
考虑组合意义，即把 n 分为 k 个集合，包含元素 1 的集合大小之和。由于所有元素等价，于是可以转为算包含每个点的大小之和，最后除 n。
那么相当于对于一个划分，它的贡献是所有集合大小的平方的和。而再根据组合意义等价于一个集合内要选择两个位置的方案数。

CF961G

首先可以直接写出答案的式子

$$\sum_{i=1}^{n} w_i \sum_{j=1}^{n} j \cdot \binom{n-1}{j-1} \left\{ {n-j \atop k-1} \right\}$$

关键在于求后面那一坨。
考虑组合意义，即把 n 分为 k 个集合，包含元素 1 的集合大小之和。由于所有元素等价，于是可以转为算包含每个点的大小之和，最后除 n。
那么相当于对于一个划分，它的贡献是所有集合大小的平方的和。而再根据组合意义等价于一个集合内要选择两个位置的方案数。
考虑假设选择的两个位置相同，那么任何划分都是合法的，故贡献是 $n \cdot \left\{ {n \atop k} \right\}$。
如果不同，那么我们认为它们相同，再套用相同的，即 $n(n-1) \cdot \left\{ {n \atop k} \right\}$

CF932E

给定 n, k，求出

$$\sum_{i=1}^{n} \binom{n}{i} \times i^k$$

$n \le 10^9$，~~$k \le 5000$~~，$k \le 10^7$

CF932E

直接进行一个式子的推:

CF932E

直接进行一个式子的推：

$$\begin{aligned}\sum_{i=0}^{n}\binom{n}{i}i^k &= \sum_{i=0}^{n}\binom{n}{i}\sum_{j=1}^{k}\left\{{k \atop j}\right\}\binom{i}{j}j! \\ &= \sum_{j=1}^{k}\left\{{k \atop j}\right\}j!\sum_{i=0}^{n}\binom{n}{i}\binom{i}{j} \\ &= \sum_{j=1}^{k}\left\{{k \atop j}\right\}\binom{n}{j}j!\sum_{i=j}^{n}\binom{n-j}{i-j} \\ &= \sum_{j=1}^{k}\left\{{k \atop j}\right\}\binom{n}{j}j!\sum_{i=0}^{n-j}\binom{n-j}{i} \\ &= \sum_{j=1}^{k}\left\{{k \atop j}\right\}\binom{n}{j}j!\cdot 2^{n-j}\end{aligned}$$

CF932E

继续推

CF932E

继续推

$$
\begin{aligned}
\sum_{j=1}^{k}\left\{{k \atop j}\right\}\binom{n}{j}j!\cdot 2^{n-j} &= \sum_{j=1}^{k}\binom{n}{j}2^{n-j}\sum_{i=0}^{j}\binom{j}{i}(-1)^{j-i}\cdot i^n \\
&= \sum_{i=0}^{k} i^n \sum_{j=i}^{k}\binom{n}{j}\binom{j}{i}(-1)^{j-i}2^{n-j} \\
&= \sum_{i=0}^{k}\binom{n}{i} i^n \sum_{j=i}^{k}\binom{n-i}{j-i}(-1)^{j-i}2^{n-j} \\
&= \sum_{i=0}^{k}\binom{n}{i} i^n 2^{n-i} \sum_{j=i}^{k}\binom{n-i}{j-i}(-1)^{j-i}2^{-(j-i)} \\
&= \sum_{i=0}^{k}\binom{n}{i} i^n 2^{n-i} \sum_{j=0}^{k-i}\binom{n-i}{j}(-1/2)^{j}
\end{aligned}
$$

CF932E

关键还是在于求后面那一坨，看起来没好方法了？注意到这种求一行组合数的形式的求和有一个很有用的 trick：递推。

CF932E

关键还是在于求后面那一坨，看起来没好方法了？注意到这种求一行组合数的形式的求和有一个很有用的 trick：递推。
我们令 f_i 表示后面的值，令 $w = -1/2$，那么要观察相邻两项差了什么，即

CF932E

关键还是在于求后面那一坨，看起来没好方法了？注意到这种求一行组合数的形式的求和有一个很有用的 trick：递推。
我们令 f_i 表示后面的值，令 $w=-1/2$，那么要观察相邻两项差了什么，即

$$
\begin{aligned}
f_i - f_{i+1} &= \sum_{j=0}^{k-i}\binom{n-i}{j}w^j - \sum_{j=0}^{k-i-1}\binom{n-i-1}{j}w^j \\
&= \binom{n-i}{k-i}w^{k-i} + \sum_{j=0}^{k-i-1}\left(\binom{n-i}{j}-\binom{n-i-1}{j}\right)w^j \\
&= \binom{n-i}{k-i}w^{k-i} + w\sum_{j=0}^{k-i-2}\binom{n-i-1}{j}w^j \\
&= \binom{n-i}{k-i}w^{k-i} + w\left(f_{i+1} - \binom{n-i-1}{k-i-1}w^{k-i-1}\right) \\
&= wf_{i+1} + w^{k-i}\binom{n-i-1}{k-i}
\end{aligned}
$$

CF932E

于是有

$$f_i = (w+1)f_{i+1} + w^{k-i}\binom{n-i-1}{k-i}$$

需要线性筛求 i^n，以及特判 $k > n$ 等细节。时间复杂度 $\mathrm{O}(k)$

P4827 Crash 的文明世界

给定一棵 n 个点的数以及常数 k，定义 $S(i) = \sum_{j=1}^{n} \operatorname{dist}(i,j)^k$。

请你对 $i = 1, 2, \ldots, n$ 求出 $S(i)$。

$n \leq 5 \times 10^4$，$k \leq 150$

P4827 Crash 的文明世界

当然可以直接树上 dp，这样的话转移需要拆二项式定理，复杂度至少为 $O(nk^2)$ 了。

P4827 Crash 的文明世界

当然可以直接树上 dp，这样的话转移需要拆二项式定理，复杂度至少为 $O(nk^2)$ 了。
使用斯特林数转组合数，于是

$$\begin{aligned} S(i) &= \sum_{j=1}^{n} \sum_{p=1}^{k} \left\{ {k \atop p} \right\} \binom{\mathrm{dist}(i,j)}{p} p! \\ &= \sum_{p=1}^{k} \left\{ {k \atop p} \right\} p! \sum_{j=1}^{n} \binom{\mathrm{dist}(i,j)}{p} \end{aligned}$$

P4827 Crash 的文明世界

当然可以直接树上 dp，这样的话转移需要拆二项式定理，复杂度至少为 $\mathrm{O}(nk^2)$ 了。
使用斯特林数转组合数，于是

$$
\begin{aligned}
S(i) &= \sum_{j=1}^{n}\sum_{p=1}^{k}\begin{Bmatrix}k\\p\end{Bmatrix}\binom{\mathrm{dist}(i,j)}{p}p! \\
&= \sum_{p=1}^{k}\begin{Bmatrix}k\\p\end{Bmatrix}p!\sum_{j=1}^{n}\binom{\mathrm{dist}(i,j)}{p}
\end{aligned}
$$

关键在于求后面那一坨，考虑组合意义，相当于所有点到 i 的路径上选择 p 条边的方案数，于是可以 dp，设 $f_{u,j}$ 表示以 u 为根的子树，选了 j 条边的方案数，这样转移是 $\mathrm{O}(1)$ 的，时间复杂度 $\mathrm{O}(nk)$，还需要一个换根 dp。

TopCoder 13444

有一个 $n \times m$ 的矩阵，每个位置填 $[1, c]$ 中的整数。
求有多少矩阵满足任意两行不完全相同且任意两列不完全相同。

$n, m \leq 4000$

TopCoder 13444

考虑如果只要求行是好做的，令 $f(n,m)$ 表示行不完全相同的方案数，那么 $f(n,m)=(c^m)^{\underline{n}}$。

TopCoder 13444

考虑如果只要求行是好做的，令 $f(n,m)$ 表示行不完全相同的方案数，那么 $f(n,m)=(c^m)^{\underline{n}}$。
令 $g(n,m)$ 表示行列都满足的方案数，那么有

$$f(n,m)=\sum_{i=1}^{m}\left\{ m \atop i \right\} g(n,i)$$

TopCoder 13444

考虑如果只要求行是好做的，令 $f(n,m)$ 表示行不完全相同的方案数，那么 $f(n,m)=(c^m)^{\underline{n}}$。
令 $g(n,m)$ 表示行列都满足的方案数，那么有

$$f(n,m)=\sum_{i=1}^{m}\begin{Bmatrix}m\\i\end{Bmatrix}g(n,i)$$

根据斯特林反演（容斥）有

$$g(n,m)=\sum_{i=1}^{m}(-1)^{m-i}\begin{bmatrix}m\\i\end{bmatrix}f(n,i)$$

总结与练习题

总结与练习题

- P6620 [省选联考 2020 A 卷] 组合数问题
- CF1278F Cards
- CF960G Bandit Blues
- 清华集训 2016 如何优雅地求和