概率期望与组合数学基础

简单计数

李嘉政[redacted]

[redacted]

2022.7.29

闲话

计数类问题是常见的，但要讲基础的东西感觉就很难了，毕竟也不知道什么是基础。

最后决定是想到什么就讲什么。

内容经过多次删减，将一些感觉比较难的内容全部放在拓展里了，感兴趣可以去学习。

课件里的习题难度比较大，刚开始只做 vjudge 上已经提供的就行，以后有时间可以做课件上的。

约定

- $[n]$ 表示 $\{1, 2, \cdots, n\}$.

约定

- $[n]$ 表示 $\{1, 2, \cdots, n\}$.
- 当条件 cond 为真时，$[\text{cond}] = 1$，否则 $[\text{cond}] = 0$.

约定

- $[n]$ 表示 $\{1, 2, \cdots, n\}$.
- 当条件 cond 为真时，$[\text{cond}] = 1$，否则 $[\text{cond}] = 0$.
- 不加说明时所有运算 $\bmod P = 998244353$.

目录

目录

加法原理

如果 $S = A \uplus B$（不交并），那么有
$\sum_{x\in S} f(x) = (\sum_{x\in A} f(x)) + (\sum_{x\in B} f(x))$.

另一种形式，即
$\sum_{x\in S}(f(x) + g(x)) = (\sum_{x\in S} f(x)) + (\sum_{x\in S} g(x))$.

期望的线性性：$E(X + Y) = E(X) + E(Y)$.

乘法原理

$$\sum_{x\in A,y\in B} f(x)g(y) = \sum_{x\in A} f(x)(\sum_{y\in B} g(y))$$
$$= (\sum_{x\in A} f(x))(\sum_{y\in B} g(y)).$$

更一般地，有 $\sum_{x_1,\cdots x_n} \prod_{i=1}^{n} f_i(x_i) = \prod_{i=1}^{n}(\sum_{x_i} f_i(x_i))$.

可以发现就是小学时学的结合律分配律。

小例题

求 $\sum_{i=1}^{n}\sum_{j=1}^{m}(i+j)$.

小例题

求 $\sum_{i=1}^{n}\sum_{j=1}^{m}(i+j)$.

直接化简。

$$
\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{m}(i+j)\\
&=\sum_{i=1}^{n}\sum_{j=1}^{m}i+\sum_{i=1}^{n}\sum_{j=1}^{m}j\\
&=(\sum_{i=1}^{n}i)(\sum_{j=1}^{m}1)+(\sum_{i=1}^{n}1)(\sum_{j=1}^{m}j).
\end{aligned}
$$

又一道小例题

给出序列 $a_1, a_2, \cdots, a_n$，求 $\sum_{1\le i\le j\le n} \bigoplus_{i\le k\le j} a_k$.
$n \le 10^5, a_i \in [0, 10^9]$.

又一道小例题

给出序列 $a_1, a_2, \cdots, a_n$，求 $\sum_{1 \le i \le j \le n} \bigoplus_{i \le k \le j} a_k$.
$n \le 10^5, a_i \in [0, 10^9]$.

- 拆位，即 $a_i = \sum_{j=0}^{30} a_{i,j} \cdot 2^j$.

又一道小例题

给出序列 $a_1, a_2, \cdots, a_n$，求 $\sum_{1\le i\le j\le n} \bigoplus_{i\le k\le j} a_k$.
$n \le 10^5, a_i \in [0, 10^9]$.

- 拆位，即 $a_i = \sum_{j=0}^{30} a_{i,j} \cdot 2^j$.
- 代入，则有 $= \sum_{1\le i\le j\le n} \bigoplus_{i\le k\le j} \sum_{l=0}^{30} a_{k,l} \cdot 2^l$.

又一道小例题

给出序列 $a_1, a_2, \cdots, a_n$，求 $\sum_{1\le i\le j\le n} \bigoplus_{i\le k\le j} a_k$.
$n \le 10^5, a_i \in [0, 10^9]$.

- 拆位，即 $a_i = \sum_{j=0}^{30} a_{i,j} \cdot 2^j$.
- 代入，则有 $= \sum_{1\le i\le j\le n} \bigoplus_{i\le k\le j} \sum_{l=0}^{30} a_{k,l} \cdot 2^l$.
- 交换求和顺序，于是 $= \sum_{l=0}^{30} 2^l (\sum_{1\le i\le j\le n} (\bigoplus_{i\le k\le j} a_{k,l}))$.
- 于是就可以 $O(n \log A)$ 求出答案。

目录

离散型随机变量

在竞赛中，遇到的绝大多数都是离散的，连续的很不常见。下面给出离散型的期望定义:

$E(X)=\sum_{x\in I(X)} xP(X=x)$.

实际上，这个定义并不优秀，想了解更自然的定义手段，可以学习勒贝格测度相关知识或阅读书籍《Probability: Theory and Examples - Rick Durrett》等。

不过这个定义已经足够我们在竞赛里证明许多东西了。

全概率公式与全期望公式

全概率公式：若事件 $A_1, A_2, \cdots, A_n$ 满足 $\forall\ i, j, A_i \cap A_j = \emptyset$ 且 $\sum_{i=1}^n P(A_i) = 1$，则有 $P(B) = \sum_{i=1}^n P(A_i)P(B|A_i)$，其中 $P(B|A)$ 表示条件概率，即在 A 已经发生的前提下，B 发生的概率，有公式 $P(B|A) = \frac{P(AB)}{P(A)}$，$AB$ 表示同时发生。

全期望公式：$E(Y) = \sum_{x \in I(X)} P(X = x)E(Y|(X = x))$，其中 $E(Y|A)$ 表示条件期望，定义与条件概率同理，$E(Y|(X = x)) = \sum_y y \frac{P(Y=y,X=x)}{P(X=x)}$.

UVa 11021 Tribles

题意：开始有 k 只麻球，每只活一天就会死亡，临死之前可能会生出一些新的麻球，最多生 n 只，生 i 只的概率为 P_i。给定 m，问 m 天后所有麻球均死亡的概率。$n, k, m \leq 10^3$.

UVa 11021 Tribles

题意：开始有 k 只麻球，每只活一天就会死亡，临死之前可能会生出一些新的麻球，最多生 n 只，生 i 只的概率为 P_i。给定 m，问 m 天后所有麻球均死亡的概率。$n, k, m \leq 10^3$.

- 注意到开始的麻球都是独立的，所以只需要先考虑一只，最后 k 次幂。

UVa 11021 Tribles

题意：开始有 k 只麻球，每只活一天就会死亡，临死之前可能会生出一些新的麻球，最多生 n 只，生 i 只的概率为 P_i。给定 m，问 m 天后所有麻球均死亡的概率。$n, k, m \leq 10^3$.

- 注意到开始的麻球都是独立的，所以只需要先考虑一只，最后 k 次幂。
- 设 f_i 表示开始只有一只麻球，i 天后没有存活麻球的概率。

UVa 11021 Tribles

题意：开始有 k 只麻球，每只活一天就会死亡，临死之前可能会生出一些新的麻球，最多生 n 只，生 i 只的概率为 P_i。给定 m，问 m 天后所有麻球均死亡的概率。$n, k, m \leq 10^3$.

- 注意到开始的麻球都是独立的，所以只需要先考虑一只，最后 k 次幂。
- 设 f_i 表示开始只有一只麻球，i 天后没有存活麻球的概率。
- 考虑第一只麻球生了 j 只麻球，此后，这 j 只麻球独立了。

UVa 11021 Tribles

题意：开始有 k 只麻球，每只活一天就会死亡，临死之前可能会生出一些新的麻球，最多生 n 只，生 i 只的概率为 P_i。给定 m，问 m 天后所有麻球均死亡的概率。$n, k, m \leq 10^3$.

- 注意到开始的麻球都是独立的，所以只需要先考虑一只，最后 k 次幂。
- 设 f_i 表示开始只有一只麻球，i 天后没有存活麻球的概率。
- 考虑第一只麻球生了 j 只麻球，此后，这 j 只麻球独立了。
- 分别考虑它们在 i 天后已经全部死亡，此时对它们而言是在第 $i-1$ 天，所以有 $f_i = \sum_{j=0}^{n} p_j \cdot f_{i-1}^{j}$.

UVa 11021 Tribles

题意：开始有 k 只麻球，每只活一天就会死亡，临死之前可能会生出一些新的麻球，最多生 n 只，生 i 只的概率为 P_i。给定 m，问 m 天后所有麻球均死亡的概率。$n, k, m \le 10^3$.

- 注意到开始的麻球都是独立的，所以只需要先考虑一只，最后 k 次幂。
- 设 f_i 表示开始只有一只麻球，i 天后没有存活麻球的概率。
- 考虑第一只麻球生了 j 只麻球，此后，这 j 只麻球独立了。
- 分别考虑它们在 i 天后已经全部死亡，此时对它们而言是在第 $i-1$ 天，所以有 $f_i = \sum_{j=0}^{n} p_j \cdot f_{i-1}^{j}$.
- 时间复杂度 $O(nm)$。

LG 4316 绿豆蛙的归宿

题意：给出张 n 个点 m 条边的有向无环图，起点为 1，终点为 n，每条边都有一个长度，并且从起点出发能够到达所有的点，所有的点也都能够到达终点。绿豆蛙从起点出发，走向终点。到达每一个顶点时，如果该节点有 k 条出边，绿豆蛙可以选择任意一条边离开该点，并且走向每条边的概率为 $\frac{1}{k}$。现在绿豆蛙想知道，从起点走到终点的所经过的路径总长度期望是多少？$n, m \leq 10^5$.

LG 4316 绿豆蛙的归宿

题意：给出张 n 个点 m 条边的有向无环图，起点为 1，终点为 n，每条边都有一个长度，并且从起点出发能够到达所有的点，所有的点也都能够到达终点。绿豆蛙从起点出发，走向终点。到达每一个顶点时，如果该节点有 k 条出边，绿豆蛙可以选择任意一条边离开该点，并且走向每条边的概率为 $\frac{1}{k}$。现在绿豆蛙想知道，从起点走到终点的所经过的路径总长度期望是多少？$n, m \le 10^5$.

- 按照日常的想法，我们设 f_i 表示从 1 到 i 的期望长度。

LG 4316 绿豆蛙的归宿

题意：给出张 n 个点 m 条边的有向无环图，起点为 1，终点为 n，每条边都有一个长度，并且从起点出发能够到达所有的点，所有的点也都能够到达终点。绿豆蛙从起点出发，走向终点。到达每一个顶点时，如果该节点有 k 条出边，绿豆蛙可以选择任意一条边离开该点，并且走向每条边的概率为 $\frac{1}{k}$。现在绿豆蛙想知道，从起点走到终点的所经过的路径总长度期望是多少？$n, m \leq 10^5$.

- 按照日常的想法，我们设 f_i 表示从 1 到 i 的期望长度。
- 考虑当前到了 x 节点，x 需要从其他节点 y 走过来，我们试着写一下转移式。

LG 4316 绿豆蛙的归宿

题意：给出张 n 个点 m 条边的有向无环图，起点为 1，终点为 n，每条边都有一个长度，并且从起点出发能够到达所有的点，所有的点也都能够到达终点。绿豆蛙从起点出发，走向终点。到达每一个顶点时，如果该节点有 k 条出边，绿豆蛙可以选择任意一条边离开该点，并且走向每条边的概率为 $\frac{1}{k}$。现在绿豆蛙想知道，从起点走到终点的所经过的路径总长度期望是多少？$n, m \leq 10^5$.

- 按照日常的想法，我们设 f_i 表示从 1 到 i 的期望长度。
- 考虑当前到了 x 节点，x 需要从其他节点 y 走过来，我们试着写一下转移式。
- $f_x = \sum_y \frac{f_y + w_{xy}}{out_y}$.

LG 4316 绿豆蛙的归宿

题意：给出张 n 个点 m 条边的有向无环图，起点为 1，终点为 n，每条边都有一个长度，并且从起点出发能够到达所有的点，所有的点也都能够到达终点。绿豆蛙从起点出发，走向终点。到达每一个顶点时，如果该节点有 k 条出边，绿豆蛙可以选择任意一条边离开该点，并且走向每条边的概率为 $\frac{1}{k}$。现在绿豆蛙想知道，从起点走到终点的所经过的路径总长度期望是多少？$n, m \le 10^5$.

- 按照日常的想法，我们设 f_i 表示从 1 到 i 的期望长度。
- 考虑当前到了 x 节点，x 需要从其他节点 y 走过来，我们试着写一下转移式。
- $f_x = \sum_y \frac{f_y + w_{xy}}{out_y}$.
- ?

LG 4316 绿豆蛙的归宿

题意：给出张 n 个点 m 条边的有向无环图，起点为 1，终点为 n，每条边都有一个长度，并且从起点出发能够到达所有的点，所有的点也都能够到达终点。绿豆蛙从起点出发，走向终点。到达每一个顶点时，如果该节点有 k 条出边，绿豆蛙可以选择任意一条边离开该点，并且走向每条边的概率为 $\frac{1}{k}$。现在绿豆蛙想知道，从起点走到终点的所经过的路径总长度期望是多少？$n, m \leq 10^5$.

- 按照日常的想法，我们设 f_i 表示从 1 到 i 的期望长度。
- 考虑当前到了 x 节点，x 需要从其他节点 y 走过来，我们试着写一下转移式。
- $f_x = \sum_y \frac{f_y + w_{xy}}{out_y}$.
- ?
- 好像和期望定义不符合。

LG 4316 绿豆蛙的归宿

- 按照日常的想法，我们设 f_i 表示从 1 到 i 的期望长度。
- 考虑当前到了 x 节点，x 需要从其他节点 y 走过来，我们试着写一下转移式。
- $f_x = \sum_y \frac{f_y + w_{xy}}{out_y}$.
- ?
- 好像和期望定义不符合。

LG 4316 绿豆蛙的归宿

- 按照日常的想法，我们设 f_i 表示从 1 到 i 的期望长度。
- 考虑当前到了 x 节点，x 需要从其他节点 y 走过来，我们试着写一下转移式。
- $f_x = \sum_y \frac{f_y + w_{xy}}{out_y}$.
- ?
- 好像和期望定义不符合。
- 尝试使用定义。

LG 4316 绿豆蛙的归宿

- 按照日常的想法，我们设 f_i 表示从 1 到 i 的期望长度。
- 考虑当前到了 x 节点，x 需要从其他节点 y 走过来，我们试着写一下转移式。
- $f_x = \sum_y \frac{f_y + w_{xy}}{out_y}$.
- ?
- 好像和期望定义不符合。
- 尝试使用定义。
- $E(y) = \sum_i i \cdot p_i$.
- $E(x|y) = \frac{\sum_i (i + w_{xy}) \cdot p_i}{out_y \cdot P(y)}$.

LG 4316 绿豆蛙的归宿

- 按照日常的想法，我们设 f_i 表示从 1 到 i 的期望长度。
- 考虑当前到了 x 节点，x 需要从其他节点 y 走过来，我们试着写一下转移式。
- $f_x = \sum_y \frac{f_y + w_{xy}}{out_y}$.
- ?
- 好像和期望定义不符合。
- 尝试使用定义。
- $E(y) = \sum_i i \cdot p_i$.
- $E(x|y) = \frac{\sum_i (i + w_{xy}) \cdot p_i}{out_y \cdot P(y)}$.
- $E(x|y) = \frac{\sum_i i \cdot p_i + \sum_i w_{xy} \cdot p_i}{out_y \cdot P(y)}$.

LG 4316 绿豆蛙的归宿

- 按照日常的想法，我们设 f_i 表示从 1 到 i 的期望长度。
- 考虑当前到了 x 节点，x 需要从其他节点 y 走过来，我们试着写一下转移式。
- $f_x = \sum_y \frac{f_y + w_{xy}}{out_y}$.
- ?
- 好像和期望定义不符合。
- 尝试使用定义。
- $E(y) = \sum_i i \cdot p_i$.
- $E(x|y) = \frac{\sum_i (i + w_{xy}) \cdot p_i}{out_y \cdot P(y)}$.
- $E(x|y) = \frac{\sum_i i \cdot p_i + \sum_i w_{xy} \cdot p_i}{out_y \cdot P(y)}$.
- $E(x|y) = \frac{E(y) + w_{xy} \sum_i p_i}{out_y \cdot P(y)}$.

LG 4316 绿豆蛙的归宿

- 按照日常的想法，我们设 f_i 表示从 1 到 i 的期望长度。
- 考虑当前到了 x 节点，x 需要从其他节点 y 走过来，我们试着写一下转移式。
- $f_x = \sum_y \frac{f_y + w_{xy}}{out_y}$.
- ?
- 好像和期望定义不符合。
- 尝试使用定义。
- $E(y) = \sum_i i \cdot p_i$.
- $E(x|y) = \frac{\sum_i (i + w_{xy}) \cdot p_i}{out_y \cdot P(y)}$.
- $E(x|y) = \frac{\sum_i i \cdot p_i + \sum_i w_{xy} \cdot p_i}{out_y \cdot P(y)}$.
- $E(x|y) = \frac{E(y) + w_{xy} \sum_i p_i}{out_y \cdot P(y)}$.
- 然后考虑全期望公式，发现前面的转移式缺少了 $\sum_i p_i$ 这一部分。

LG 4316 绿豆蛙的归宿

- 想一想 $\sum_i p_i$ 是啥。

LG 4316 绿豆蛙的归宿

- 想一想 $\sum_i p_i$ 是啥。
- 正是 1 到 y 的概率，这可以直接递推出来，设为 g_y.

LG 4316 绿豆蛙的归宿

- 想一想 $\sum_i p_i$ 是啥。
- 正是 1 到 y 的概率，这可以直接递推出来，设为 g_y.
- 于是成功得到了，$f_x = \sum_y \frac{f_y + w_{xy} \cdot g_y}{out_y}$.

LG 4316 绿豆蛙的归宿

- 想一想 $\sum_i p_i$ 是啥。
- 正是 1 到 y 的概率，这可以直接递推出来，设为 g_y.
- 于是成功得到了，$f_x = \sum_y \frac{f_y + w_{xy} \cdot g_y}{out_y}$.
- 时间复杂度 $O(n+m)$，拓扑序递推即可。

LG 4316 绿豆蛙的归宿

- 有没有更简单的想法?

LG 4316 绿豆蛙的归宿

- 有没有更简单的想法?
- 注意到我们最终要求的是 1 到 n 的期望长度，所以还可以设 f_x 表示 x 到 n 的期望长度。

LG 4316 绿豆蛙的归宿

- 有没有更简单的想法?
- 注意到我们最终要求的是 1 到 n 的期望长度，所以还可以设 f_x 表示 x 到 n 的期望长度。
- 发现递推式异常简单，是 $f_x = \sum_y \frac{f_y + w_{xy}}{out_x}$，只不过此时的 y 是 x 将要走到的节点。

LG 4316 绿豆蛙的归宿

- 有没有更简单的想法?
- 注意到我们最终要求的是 1 到 n 的期望长度，所以还可以设 f_x 表示 x 到 n 的期望长度。
- 发现递推式异常简单，是 $f_x = \sum_y \frac{f_y + w_{xy}}{out_x}$，只不过此时的 y 是 x 将要走到的节点。
- 为什么会这样?

LG 4316 绿豆蛙的归宿

- 有没有更简单的想法?
- 注意到我们最终要求的是 1 到 n 的期望长度，所以还可以设 f_x 表示 x 到 n 的期望长度。
- 发现递推式异常简单，是 $f_x = \sum_y \frac{f_y + w_{xy}}{out_x}$，只不过此时的 y 是 x 将要走到的节点。
- 为什么会这样?
- 原因很简单，上面我们用到了 g_x，而此时 $g_x = 1$，因为终点只有一个。

LG 4316 绿豆蛙的归宿

- 有没有更简单的想法?
- 注意到我们最终要求的是 1 到 n 的期望长度，所以还可以设 f_x 表示 x 到 n 的期望长度。
- 发现递推式异常简单，是 $f_x = \sum_y \frac{f_y + w_{xy}}{out_x}$，只不过此时的 y 是 x 将要走到的节点。
- 为什么会这样?
- 原因很简单，上面我们用到了 g_x，而此时 $g_x = 1$，因为终点只有一个。
- 算期望的题中，终点只有一个非常常见，所以这样的倒推思路是最常见的。

HUSTPC2022 for test H Escaping from Monsters

题意：有 n 只怪兽，第 i 只的防御力为 a_i，你需要按序去杀死每一只怪兽。你的初始值攻击力 m 为 0，假设你当前杀到了第 i 只怪兽，若 $m = a_i$，你直接杀死了怪兽且没有损耗；若 $m > a_i$，你成功杀死了怪兽，但有 p_i 的概率让攻击力变成 $m-1$；若 $m < a_i$，则你杀不死怪兽，但你有 p_i 的概率使得你的攻击力变成 $m+1$。问期望战斗轮数。(无论是否杀死怪兽都算一轮战斗)

$n, a_i \leq 10^3$.

HUSTPC2022 for test H Escaping from Monsters

题意：有 n 只怪兽，第 i 只的防御力为 a_i，你需要按序去杀死每一只怪兽。你的初始值攻击力 m 为 0，假设你当前杀到了第 i 只怪兽，若 $m = a_i$，你直接杀死了怪兽且没有损耗；若 $m > a_i$，你成功杀死了怪兽，但有 p_i 的概率让攻击力变成 $m-1$；若 $m < a_i$，则你杀不死怪兽，但你有 p_i 的概率使得你的攻击力变成 $m+1$。问期望战斗轮数。(无论是否杀死怪兽都算一轮战斗)
$n, a_i \leq 10^3$.

- 跟上题一致，我们有正推和倒推的想法。

HUSTPC2022 for test H Escaping from Monsters

题意：有 n 只怪兽，第 i 只的防御力为 a_i，你需要按序去杀死每一只怪兽。你的初始值攻击力 m 为 0，假设你当前杀到了第 i 只怪兽，若 $m = a_i$，你直接杀死了怪兽且没有损耗；若 $m > a_i$，你成功杀死了怪兽，但有 p_i 的概率让攻击力变成 $m-1$；若 $m < a_i$，则你杀不死怪兽，但你有 p_i 的概率使得你的攻击力变成 $m+1$。问期望战斗轮数。(无论是否杀死怪兽都算一轮战斗)
$n, a_i \leq 10^3$.

- 跟上题一致，我们有正推和倒推的想法。
- 容易发现倒推比较简单，所以采取倒推。

HUSTPC2022 for test H Escaping from Monsters

题意：有 n 只怪兽，第 i 只的防御力为 a_i，你需要按序去杀死每一只怪兽。你的初始值攻击力 m 为 0，假设你当前杀到了第 i 只怪兽，若 $m = a_i$，你直接杀死了怪兽且没有损耗；若 $m > a_i$，你成功杀死了怪兽，但有 p_i 的概率让攻击力变成 $m-1$；若 $m < a_i$，则你杀不死怪兽，但你有 p_i 的概率使得你的攻击力变成 $m+1$。问期望战斗轮数。(无论是否杀死怪兽都算一轮战斗)
$n, a_i \le 10^3$.

- 跟上题一致，我们有正推和倒推的想法。
- 容易发现倒推比较简单，所以采取倒推。
- 设 $f_{i,j}$ 表示从第 i 个怪打到第 n 个怪，当前我的攻击力为 j 的期望轮数。

HUSTPC2022 for test H Escaping from Monsters

题意：有 n 只怪兽，第 i 只的防御力为 a_i，你需要按序去杀死每一只怪兽。你的初始值攻击力 m 为 0，假设你当前杀到了第 i 只怪兽，若 $m = a_i$，你直接杀死了怪兽且没有损耗；若 $m > a_i$，你成功杀死了怪兽，但有 p_i 的概率让攻击力变成 $m-1$；若 $m < a_i$，则你杀不死怪兽，但你有 p_i 的概率使得你的攻击力变成 $m+1$。问期望战斗轮数。(无论是否杀死怪兽都算一轮战斗)
$n, a_i \leq 10^3$.

- 跟上题一致，我们有正推和倒推的想法。
- 容易发现倒推比较简单，所以采取倒推。
- 设 $f_{i,j}$ 表示从第 i 个怪打到第 n 个怪，当前我的攻击力为 j 的期望轮数。
- $j \geq a_i$ 时转移很显然，但如果小于呢?

HUSTPC2022 for test H Escaping from Monsters

- 跟上题一致，我们有正推和倒推的想法。
- 容易发现倒推比较简单，所以采取倒推。
- 设 $f_{i,j}$ 表示从第 i 个怪打到第 n 个怪，当前我的攻击力为 j 的期望轮数。
- $j \geq a_i$ 时转移很显然，但如果小于呢?
- 等价从 0 走到 m，每次有 p 的概率成功，问期望步数。

HUSTPC2022 for test H Escaping from Monsters

- 跟上题一致，我们有正推和倒推的想法。
- 容易发现倒推比较简单，所以采取倒推。
- 设 $f_{i,j}$ 表示从第 i 个怪打到第 n 个怪，当前我的攻击力为 j 的期望轮数。
- $j \geq a_i$ 时转移很显然，但如果小于呢?
- 等价从 0 走到 m，每次有 p 的概率成功，问期望步数。
- 设 g_m 表示答案，显然有转移 $g_m = (1-p) \cdot g_m + p \cdot g_{m-1} + 1.$

HUSTPC2022 for test H Escaping from Monsters

- 跟上题一致，我们有正推和倒推的想法。
- 容易发现倒推比较简单，所以采取倒推。
- 设 $f_{i,j}$ 表示从第 i 个怪打到第 n 个怪，当前我的攻击力为 j 的期望轮数。
- $j \geq a_i$ 时转移很显然，但如果小于呢?
- 等价从 0 走到 m，每次有 p 的概率成功，问期望步数。
- 设 g_m 表示答案，显然有转移 $g_m = (1-p) \cdot g_m + p \cdot g_{m-1} + 1.$
- 直接手动解出了 g_m，发现这其实是带环的动态规划，后面会提到。

HUSTPC2022 for test H Escaping from Monsters

- 跟上题一致，我们有正推和倒推的想法。
- 容易发现倒推比较简单，所以采取倒推。
- 设 $f_{i,j}$ 表示从第 i 个怪打到第 n 个怪，当前我的攻击力为 j 的期望轮数。
- $j \geq a_i$ 时转移很显然，但如果小于呢?
- 等价从 0 走到 m，每次有 p 的概率成功，问期望步数。
- 设 g_m 表示答案，显然有转移
  $g_m = (1-p) \cdot g_m + p \cdot g_{m-1} + 1.$
- 直接手动解出了 g_m，发现这其实是带环的动态规划，后面会提到。
- 这题其实也可以正推，如何做自己思考。提示一下，可以直接算概率。

期望与高斯消元

在上一题中，我们就遇到了 $g_m = (1-p) \cdot g_m + p \cdot g_{m-1} + 1$ 这样的东西，容易发现这东西会从自己转移到自己的，又或者说是形成了环。一般情况下，遇到这种东西，我们会考虑把它看做线性方程组，然后用我们线代里熟知的高斯消元去解决，但实际上，很多题目不需要真的去高斯消元，而可以用其他办法来优化高斯消元的效率。

loj 6118 鬼牌

题意：小鬼找来了 m 种牌，大小分别为 1 到 m，其中大小为 i 的牌有 a_i 张，他会使用鬼的能力改变牌的大小，但由于小鬼很不熟练，他每次操作实际上可以看做随机选两张牌 A 和 B，将牌 A 的大小变为与牌 B 一致。当所有的牌大小都一致时，它们就组成了一个很大的炸弹，小鬼想知道他操作数目的期望值。
$\sum_{i=1}^{m} a_i \leq 10^9; m \leq 10^5$.

loj 6118 鬼牌

题意：小鬼找来了 m 种牌，大小分别为 1 到 m，其中大小为 i 的牌有 a_i 张，他会使用鬼的能力改变牌的大小，但由于小鬼很不熟练，他每次操作实际上可以看做随机选两张牌 A 和 B，将牌 A 的大小变为与牌 B 一致。当所有的牌大小都一致时，它们就组成了一个很大的炸弹，小鬼想知道他操作数目的期望值。$\sum_{i=1}^{m} a_i \leq 10^9; m \leq 10^5$.

- 考虑枚举最后一张牌是什么，那么要么全变成这张牌，要么这张牌最后全部没了，这样我们就只用把所有牌的期望值加起来即可，这里用到了期望的线性性。

loj 6118 鬼牌

题意：小鬼找来了 m 种牌，大小分别为 1 到 m，其中大小为 i 的牌有 a_i 张，他会使用鬼的能力改变牌的大小，但由于小鬼很不熟练，他每次操作实际上可以看做随机选两张牌 A 和 B，将牌 A 的大小变为与牌 B 一致。当所有的牌大小都一致时，它们就组成了一个很大的炸弹，小鬼想知道他操作数目的期望值。$\sum_{i=1}^{m} a_i \leq 10^9; m \leq 10^5$.

- 考虑枚举最后一张牌是什么，那么要么全变成这张牌，要么这张牌最后全部没了，这样我们就只用把所有牌的期望值加起来即可，这里用到了期望的线性性。
- 于是问题变成了，有两种牌，一种是 i 张，一种是 $n-i$ 张，问全变成第一张的期望操作数。我们设答案为 f_i。

loj 6118 鬼牌

题意：小鬼找来了 m 种牌，大小分别为 1 到 m，其中大小为 i 的牌有 a_i 张，他会使用鬼的能力改变牌的大小，但由于小鬼很不熟练，他每次操作实际上可以看做随机选两张牌 A 和 B，将牌 A 的大小变为与牌 B 一致。当所有的牌大小都一致时，它们就组成了一个很大的炸弹，小鬼想知道他操作数目的期望值。$\sum_{i=1}^{m} a_i \le 10^9; m \le 10^5$.

- 考虑枚举最后一张牌是什么，那么要么全变成这张牌，要么这张牌最后全部没了，这样我们就只用把所有牌的期望值加起来即可，这里用到了期望的线性性。
- 于是问题变成了，有两种牌，一种是 i 张，一种是 $n-i$ 张，问全变成第一张的期望操作数。我们设答案为 f_i。
- 在列转移式时，我们发现了两个问题，一是这个转移同时关系到 f_{i-1} 与 f_{i+1}，这样状态构成了环，不再是拓扑序。二是我们还要算从 i 到 $i-1$ 的概率。

loj 6118 鬼牌

- 设挑出一样的牌的概率为 $p = 1 - \frac{2*i*(n-i)}{n*(n-1)}$，使 i 变成 $i-1$ 的概率为 $q = \frac{i*(n-i)}{n*(n-1)}$，我们容易写出一个 $\sum$ 的式子，即 $\sum_{j\ge 0}(j+1)*p^j*q$，含义就是挑了几次一样的牌。化简得到这个期望步数为 $\frac{q}{(1-p)^2}$.

loj 6118 鬼牌

- 设挑出一样的牌的概率为 $p = 1 - \frac{2*i*(n-i)}{n*(n-1)}$，使 i 变成 $i-1$ 的概率为 $q = \frac{i*(n-i)}{n*(n-1)}$，我们容易写出一个 $\sum$ 的式子，即 $\sum_{j\ge 0}(j+1)*p^j*q$，含义就是挑了几次一样的牌。化简得到这个期望步数为 $\frac{q}{(1-p)^2}$.
- 注意到我们的 f_i 指的是全变成第一张的期望操作数，于是这里的期望步数还要乘上变成第一张的概率，设 g_i 表示两种牌，第一种 i 张，第二种 $n-i$ 张，全变成第一张的概率。

loj 6118 鬼牌

- 设挑出一样的牌的概率为 $p = 1 - \frac{2*i*(n-i)}{n*(n-1)}$，使 i 变成 $i-1$ 的概率为 $q = \frac{i*(n-i)}{n*(n-1)}$，我们容易写出一个 $\sum$ 的式子，即 $\sum_{j\ge 0}(j+1)*p^j*q$，含义就是挑了几次一样的牌。化简得到这个期望步数为 $\frac{q}{(1-p)^2}$.
- 注意到我们的 f_i 指的是全变成第一张的期望操作数，于是这里的期望步数还要乘上变成第一张的概率，设 g_i 表示两种牌，第一种 i 张，第二种 $n-i$ 张，全变成第一张的概率。
- 概率是容易的，一下子就列出转移式，$g_i = \frac{1}{2}(g_{i-1}+g_{i+1})$.

loj 6118 鬼牌

- 设挑出一样的牌的概率为 $p = 1 - \frac{2*i*(n-i)}{n*(n-1)}$，使 i 变成 $i-1$ 的概率为 $q = \frac{i*(n-i)}{n*(n-1)}$，我们容易写出一个 $\sum$ 的式子，即 $\sum_{j\geq 0}(j+1)*p^j*q$，含义就是挑了几次一样的牌。化简得到这个期望步数为 $\frac{q}{(1-p)^2}$.
- 注意到我们的 f_i 指的是全变成第一张的期望操作数，于是这里的期望步数还要乘上变成第一张的概率，设 g_i 表示两种牌，第一种 i 张，第二种 $n-i$ 张，全变成第一张的概率。
- 概率是容易的，一下子就列出转移式，$g_i = \frac{1}{2}(g_{i-1} + g_{i+1})$.
- 像高中数列题那样，写成 $g_{i+1} = 2*g_i - g_{i-1}$。数学归纳法，我们意外发现 $g_i = i \cdot g_1$，又 $g_n = 1$，于是 $g_i = \frac{i}{n}$.

loj 6118 鬼牌

- 设挑出一样的牌的概率为 $p=1-\frac{2*i*(n-i)}{n*(n-1)}$，使 i 变成 $i-1$ 的概率为 $q=\frac{i*(n-i)}{n*(n-1)}$，我们容易写出一个 $\sum$ 的式子，即 $\sum_{j\geq 0}(j+1)*p^j*q$，含义就是挑了几次一样的牌。化简得到这个期望步数为 $\frac{q}{(1-p)^2}$.
- 注意到我们的 f_i 指的是全变成第一张的期望操作数，于是这里的期望步数还要乘上变成第一张的概率，设 g_i 表示两种牌，第一种 i 张，第二种 $n-i$ 张，全变成第一张的概率。
- 概率是容易的，一下子就列出转移式，$g_i=\frac{1}{2}(g_{i-1}+g_{i+1})$.
- 像高中数列题那样，写成 $g_{i+1}=2*g_i-g_{i-1}$。数学归纳法，我们意外发现 $g_i=i\cdot g_1$，又 $g_n=1$，于是 $g_i=\frac{i}{n}$.
- 于是，转移式就是 $f_i=\frac{1}{2}(f_{i-1}+f_{i+1})+\frac{q}{(1-p)^2}\cdot(\frac{i-1}{n}+\frac{i+1}{n})$.

loj 6118 鬼牌

- 于是，转移式就是 $f_i = \frac{1}{2}(f_{i-1} + f_{i+1}) + \frac{q}{(1-p)^2} \cdot (\frac{i-1}{n} + \frac{i+1}{n})$.
- 进一步，$f_{i+1} = 2 * f_i - f_{i-1} - \frac{n-1}{n-i}$.

loj 6118 鬼牌

- 于是，转移式就是 $f_i = \frac{1}{2}(f_{i-1} + f_{i+1}) + \frac{q}{(1-p)^2} \cdot (\frac{i-1}{n} + \frac{i+1}{n})$.
- 进一步， $f_{i+1} = 2 * f_i - f_{i-1} - \frac{n-1}{n-i}$.
- 同样地，根据观察和数学归纳法以及 $f_n = 0$，我们求出 $f_i = i * \frac{(n-1)*(n-1)}{n} - i*(n-1)*(\sum_{j=1}^{i-1} \frac{1}{n-j}) + (n-1)*(\sum_{j=1}^{i-1} \frac{j}{n-j})$.

loj 6118 鬼牌

- 于是，转移式就是 $f_i = \frac{1}{2}(f_{i-1} + f_{i+1}) + \frac{q}{(1-p)^2} \cdot (\frac{i-1}{n} + \frac{i+1}{n})$.
- 进一步，$f_{i+1} = 2 * f_i - f_{i-1} - \frac{n-1}{n-i}$.
- 同样地，根据观察和数学归纳法以及 $f_n = 0$，我们求出 $f_i = i * \frac{(n-1)*(n-1)}{n} - i * (n-1) * (\sum_{j=1}^{i-1} \frac{1}{n-j}) + (n-1) * (\sum_{j=1}^{i-1} \frac{j}{n-j})$.
- 设 $H_n = \sum_{i=1}^{n} \frac{1}{i}$，经过简单的化简，可以求得 $f_i = i * \frac{(n-1)*(n-1)}{n} + (n-1) * (n-i) * (H_{n-1} - H_{n-i}) - (n-1) * (i-1)$.

loj 6118 鬼牌

- 于是，转移式就是 $f_i = \frac{1}{2}(f_{i-1} + f_{i+1}) + \frac{q}{(1-p)^2} \cdot (\frac{i-1}{n} + \frac{i+1}{n})$.
- 进一步，$f_{i+1} = 2 * f_i - f_{i-1} - \frac{n-1}{n-i}$.
- 同样地，根据观察和数学归纳法以及 $f_n = 0$，我们求出 $f_i = i * \frac{(n-1)*(n-1)}{n} - i*(n-1)*(\sum_{j=1}^{i-1} \frac{1}{n-j}) + (n-1)*(\sum_{j=1}^{i-1} \frac{j}{n-j})$.
- 设 $H_n = \sum_{i=1}^{n} \frac{1}{i}$，经过简单的化简，可以求得 $f_i = i * \frac{(n-1)*(n-1)}{n} + (n-1)*(n-i)*(H_{n-1} - H_{n-i}) - (n-1)*(i-1)$.
- 注意到 H 是调和级数，所以可以用欧拉常数近似或者分块打表等手段算出答案，时间复杂度 $O(m)$.

loj 6118 鬼牌

- 于是，转移式就是 $f_i=\frac{1}{2}(f_{i-1}+f_{i+1})+\frac{q}{(1-p)^2}\cdot(\frac{i-1}{n}+\frac{i+1}{n})$.
- 进一步，$f_{i+1}=2*f_i-f_{i-1}-\frac{n-1}{n-i}$.
- 同样地，根据观察和数学归纳法以及 $f_n=0$，我们求出 $f_i=i*\frac{(n-1)*(n-1)}{n}-i*(n-1)*(\sum_{j=1}^{i-1}\frac{1}{n-j})+(n-1)*(\sum_{j=1}^{i-1}\frac{j}{n-j})$.
- 设 $H_n=\sum_{i=1}^{n}\frac{1}{i}$，经过简单的化简，可以求得 $f_i=i*\frac{(n-1)*(n-1)}{n}+(n-1)*(n-i)*(H_{n-1}-H_{n-i})-(n-1)*(i-1)$.
- 注意到 H 是调和级数，所以可以用欧拉常数近似或者分块打表等手段算出答案，时间复杂度 $O(m)$.
- Extra: 这题其实有更简单的做法，感兴趣可以去了解离散时间鞅相关的知识。

CF 802L Send the Fool Further!

题意：给你一颗树，从 1 号点开始随机游走，如果走到叶子就停下来，问你期望步数。$n \leq 10^5$.

CF 802L Send the Fool Further!

题意：给你一颗树，从 1 号点开始随机游走，如果走到叶子就停下来，问你期望步数。$n \le 10^5$.

- 仍然采取倒序计算，设 f_x 表示从 x 开始游走到叶子的期望时间。

CF 802L Send the Fool Further!

题意：给你一颗树，从 1 号点开始随机游走，如果走到叶子就停下来，问你期望步数。$n \le 10^5$.

- 仍然采取倒序计算，设 f_x 表示从 x 开始游走到叶子的期望时间。
- 转移是显然的，即 $f_x = \frac{f_{fa}+w_{xfa}+\sum_y f_y+w_{xy}}{deg_x}$.

CF 802L Send the Fool Further!

题意：给你一颗树，从 1 号点开始随机游走，如果走到叶子就停下来，问你期望步数。$n \leq 10^5$.

- 仍然采取倒序计算，设 f_x 表示从 x 开始游走到叶子的期望时间。
- 转移是显然的，即 $f_x = \frac{f_{fa}+w_{x\,fa}+\sum_y f_y+w_{xy}}{deg_x}$.
- 又带环了，而且看起来没法直接手动消元，那怎么办呢?

CF 802L Send the Fool Further!

题意：给你一颗树，从 1 号点开始随机游走，如果走到叶子就停下来，问你期望步数。$n \le 10^5$.

- 仍然采取倒序计算，设 f_x 表示从 x 开始游走到叶子的期望时间。
- 转移是显然的，即 $f_x = \frac{f_{fa}+w_{xfa}+\sum_y f_y+w_{xy}}{deg_x}$.
- 又带环了，而且看起来没法直接手动消元，那怎么办呢?
- 注意到转移是线性的，即 f_x 是关于 f_{fa} 的一次函数。证明就数学归纳法，假设 x 子节点的 f 都是 f_x 的一次函数，则 f_x 也就是 f_{fa} 的一次函数了，归纳成立。

CF 802L Send the Fool Further!

题意：给你一颗树，从 1 号点开始随机游走，如果走到叶子就停下来，问你期望步数。$n \leq 10^5$.

- 仍然采取倒序计算，设 f_x 表示从 x 开始游走到叶子的期望时间。
- 转移是显然的，即 $f_x = \frac{f_{fa}+w_{xfa}+\sum_y f_y+w_{xy}}{deg_x}$.
- 又带环了，而且看起来没法直接手动消元，那怎么办呢？
- 注意到转移是线性的，即 f_x 是关于 f_{fa} 的一次函数。证明就数学归纳法，假设 x 子节点的 f 都是 f_x 的一次函数，则 f_x 也就是 f_{fa} 的一次函数了，归纳成立。
- 于是利用转移式，深度优先搜索整棵树，于是就能解出 $f_x = k_x f_{fa} + b_x$。特殊地，根节点没有父亲，此时转移式为 $f_1 = \frac{0+0+\sum_y f_y+w_{1y}}{deg_1}$，注意到我们此刻可以将 f_{fa} 与 w_{xfa} 看成 0，并不影响递推结果。于是 b_1 也就是答案了。时间复杂度 $O(n)$.

一些习题

NOIP2016 换教室

SCOI2008 奖励关

SHOI2012 随机树

第 46 届 ICPC 亚洲区域赛（昆明）B Blocks

loj 6357 game

LG 5637 ckw 的树

ZJOI2015 地震后的幻想乡

拓展

很多期望题会与组合计数相关，于是自然会有概率生成函数这种东西的出现，相关例题：

CTSC2006 歌唱王国

期望相关的高斯消元也具备许多特殊性，网格图的随机游走中，虽然高斯消元的矩阵是 $n^2 \times n^2$ 的，但实际上每一行有值的列都是连续的 $2n$ 个，我们称这种矩阵叫 Band Matrix，有一些特殊的消元方法，相关例题：

CF 963E Circles of Waiting

Hdu 6994 Pony Running

拓展

同时，一些矩阵的秩实际上不大，但矩阵很大。这些情况下如果我们能用一组基底表达出其他元，那消元的数量就大幅减少了，这种做法我们称为主元法，相关例题：

CF 963E Circles of Waiting

前面的一些题我们注意到了所有值可以被某个初始值线性表出，于是我们去递推一次函数来消元。如果并不能线性表出，那又有什么做法呢？一些题中，可以注意到初始值是单调的，于是可以二分初始值来计算，相关例题：

NWERC2020 G Great Expectations

更特殊地，一些题中直接具有常系数线性递推式，所以可以打表，用 BM 求出递推式，相关例题：

300iq Contest 1E Expected Value

目录

排列

- 从 n 个元素的集合中取出 m 个不同元素（$(m \leq n)$ 有序摆放的方案数 $P(n,m) = n^{\underline{m}} = n \times (n-1) \times \cdots \times (n-m+1)$，我们称 $n^{\underline{m}}$ 为下降幂。

排列

- 从 n 个元素的集合中取出 m 个不同元素（$(m \leq n)$ 有序摆放的方案数 $P(n,m) = n^{\underline{m}} = n \times (n-1) \times \cdots \times (n-m+1)$，我们称 $n^{\underline{m}}$ 为下降幂。
- n 个不同元素有序摆放 (全排列) 的方案数 $P(n,n) = n!$，称为 n 的阶乘，因此有 $P(n,m) = \frac{n!}{(n-m)!}$.

排列

- 从 n 个元素的集合中取出 m 个不同元素（($m \le n$) 有序摆放的方案数 $P(n,m) = n^{\underline{m}} = n \times (n-1) \times \cdots \times (n-m+1)$，我们称 $n^{\underline{m}}$ 为下降幂。
- n 个不同元素有序摆放 (全排列) 的方案数 $P(n,n) = n!$，称为 n 的阶乘，因此有 $P(n,m) = \frac{n!}{(n-m)!}$.
- n 个不同元素有序地排成一个环的方案数为 $\frac{n!}{n} = (n-1)!$.

组合

- 从 n 个元素的集合中取出 m 个不同元素 $(m \le n)$ 组成集合的方案数 $C(n,m) = \frac{n\times(n-1)\times\cdots\times(n-m+1)}{m!} = \frac{n!}{m!(n-m)!}$，这条式子同时给出了一种求组合数的方式，即预处理 $n!$ 与 $n!$ 的逆元。

组合

- 从 n 个元素的集合中取出 m 个不同元素 $(m \le n)$ 组成集合的方案数 $C(n,m) = \frac{n\times(n-1)\times\cdots\times(n-m+1)}{m!} = \frac{n!}{m!(n-m)!}$，这条式子同时给出了一种求组合数的方式，即预处理 $n!$ 与 $n!$ 的逆元。
- 选出子集或补集等价，即 $C(n,m) = C(n,n-m)$.

组合

- 从 n 个元素的集合中取出 m 个不同元素 $(m \le n)$ 组成集合的方案数 $C(n,m) = \frac{n\times(n-1)\times\cdots\times(n-m+1)}{m!} = \frac{n!}{m!(n-m)!}$，这条式子同时给出了一种求组合数的方式，即预处理 $n!$ 与 $n!$ 的逆元。
- 选出子集或补集等价，即 $C(n,m) = C(n,n-m)$.
- 若可重集里有 m 种元素，第 i 种元素有 n_i 个，令 $n = \sum_{i=1}^{m} n_i$，则该可重集的全排列数量为 $\frac{n!}{\prod_{i=1}^{m} n_i!}$.

二项式系数

二项式定理：

$$(x+y)^n = \sum_{k=-\infty}^{\infty} \binom{n}{k} x^{n-k} y^k.$$

其中，我们称 $\binom{n}{k}$ 为二项式系数，n 为任意实数，k 为整数，其定义为

$$\binom{n}{k} = \begin{cases} \dfrac{n(n-1)\cdots(n-k+1)}{k!} & k \geq 1 \\ 1 & k = 0 \\ 0 & k \leq -1 \end{cases}$$

容易发现，$C(n,k)$ 是二项式系数的特例。

二项式系数相关的恒等式

对于非负整数 n, a, b, l, r，有

- $\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m}$.

二项式系数相关的恒等式

对于非负整数 n, a, b, l, r，有

- $\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m}$.
- $\sum_{k=0}^{n} \binom{n}{k} = 2^n, \sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n$.

二项式系数相关的恒等式

对于非负整数 n, a, b, l, r，有

- $\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m}$.
- $\sum_{k=0}^{n} \binom{n}{k} = 2^n, \sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n$.
- $\sum_{k\geq 0} \binom{n}{2k} = \sum_{k\geq 0} \binom{n}{2k+1} = 2^{n-1}$.

二项式系数相关的恒等式

对于非负整数 n, a, b, l, r，有

- $\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m}$.
- $\sum_{k=0}^{n} \binom{n}{k} = 2^n, \sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n$.
- $\sum_{k \geq 0} \binom{n}{2k} = \sum_{k \geq 0} \binom{n}{2k+1} = 2^{n-1}$.
- $\sum_{k=0}^{n} k \binom{n}{k} = n 2^{n-1}$.

二项式系数相关的恒等式

对于非负整数 n, a, b, l, r，有

- $\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m}$.
- $\sum_{k=0}^{n} \binom{n}{k} = 2^n, \sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n$.
- $\sum_{k\geq 0} \binom{n}{2k} = \sum_{k\geq 0} \binom{n}{2k+1} = 2^{n-1}$.
- $\sum_{k=0}^{n} k\binom{n}{k} = n2^{n-1}$.
- $\sum_{k=l}^{r} \binom{k}{m} = \binom{r+1}{m+1} - \binom{l}{m+1}, l \leq r$.

二项式系数相关的恒等式

对于非负整数 n, a, b, l, r，有

- $\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m}$.
- $\sum_{k=0}^{n} \binom{n}{k} = 2^n, \sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n$.
- $\sum_{k \geq 0} \binom{n}{2k} = \sum_{k \geq 0} \binom{n}{2k+1} = 2^{n-1}$.
- $\sum_{k=0}^{n} k \binom{n}{k} = n2^{n-1}$.
- $\sum_{k=l}^{r} \binom{k}{m} = \binom{r+1}{m+1} - \binom{l}{m+1}, l \leq r$.
- $\sum_{k=0}^{n} \binom{a}{k} \binom{b}{n-k} = \binom{a+b}{n}$.

二项式系数相关的恒等式

对于非负整数 n, a, b, l, r，有

- $\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m}$.
- $\sum_{k=0}^{n} \binom{n}{k} = 2^n, \sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n$.
- $\sum_{k\geq 0} \binom{n}{2k} = \sum_{k\geq 0} \binom{n}{2k+1} = 2^{n-1}$.
- $\sum_{k=0}^{n} k\binom{n}{k} = n2^{n-1}$.
- $\sum_{k=l}^{r} \binom{k}{m} = \binom{r+1}{m+1} - \binom{l}{m+1}, l \leq r$.
- $\sum_{k=0}^{n} \binom{a}{k}\binom{b}{n-k} = \binom{a+b}{n}$.
- 以上的恒等式除了最后一条以外，其他的用大家已经学过的级数或者二项式定理相关知识都是容易证明的。最后一条叫范德蒙德卷积，有组合意义证明，可以自行思考。

二项式系数相关的小问题

- 从 $(0,0)$ 出发，每走一步要么横坐标加一，要么纵坐标加一，走到 (n,m) 的路径有 $\binom{n+m}{n}$ 种。

二项式系数相关的小问题

- 从 $(0,0)$ 出发，每走一步要么横坐标加一，要么纵坐标加一，走到 (n,m) 的路径有 $\binom{n+m}{n}$ 种。
- 关于 x_i 的方程 $x_1+x_2+\cdots+x_m=n$ 具有 $\binom{n+m-1}{m-1}$ 组非负整数解。

二项式系数相关的小问题

- 从 $(0,0)$ 出发，每走一步要么横坐标加一，要么纵坐标加一，走到 (n,m) 的路径有 $\binom{n+m}{n}$ 种。
- 关于 x_i 的方程 $x_1+x_2+\cdots+x_m=n$ 具有 $\binom{n+m-1}{m-1}$ 组非负整数解。
- 第二个小问题叫做隔板法，证明如下：先把所有 x_i 加一，这样的话等价于正整数之和等于 $n+m$。考虑有 $n+m$ 个小球摆成一排，现要划分出 m 段，每段至少一个小球。一共有 $n+m-1$ 的间隔，等价于这些间隔中选出 $m-1$ 个，于是就得证了。

二项式系数相关的小问题

- 从 $(0,0)$ 出发，每走一步要么横坐标加一，要么纵坐标加一，走到 (n,m) 的路径有 $\binom{n+m}{n}$ 种。
- 关于 x_i 的方程 $x_1+x_2+\cdots+x_m=n$ 具有 $\binom{n+m-1}{m-1}$ 组非负整数解。
- 第二个小问题叫做隔板法，证明如下：先把所有 x_i 加一，这样的话等价于正整数之和等于 $n+m$。考虑有 $n+m$ 个小球摆成一排，现要划分出 m 段，每段至少一个小球。一共有 $n+m-1$ 的间隔，等价于这些间隔中选出 $m-1$ 个，于是就得证了。
- 1 到 n 这 n 个自然数中选 k 个，这 k 个数任何两个数不相邻的方案数是 $\binom{n-k+1}{k}$，隔板法可以轻松证明。

SDOI2016 排列计数

题意：求有多少种 n 的排列 A，满足；若第 i 个数 A_i 的值为 i，则称 i 是稳定的。序列恰好有 m 个数是稳定的。T 组询问，$T, n, m \leq 10^6$.

SDOI2016 排列计数

题意：求有多少种 n 的排列 A，满足：若第 i 个数 A_i 的值为 i，则称 i 是稳定的。序列恰好有 m 个数是稳定的。T 组询问，$T, n, m \leq 10^6$.

- 考虑先放好 m 个位置，这些位置让它就等于 i，然后接下来都必须是不稳定的。

SDOI2016 排列计数

题意：求有多少种 n 的排列 A，满足：若第 i 个数 A_i 的值为 i，则称 i 是稳定的。序列恰好有 m 个数是稳定的。T 组询问，$T, n, m \le 10^6$.

- 考虑先放好 m 个位置，这些位置让它就等于 i，然后接下来都必须是不稳定的。
- 发现不稳定的方案等价于 $n-m$ 的错排方案数，设其为 f_{n-m}。

SDOI2016 排列计数

题意：求有多少种 n 的排列 A，满足：若第 i 个数 A_i 的值为 i，则称 i 是稳定的。序列恰好有 m 个数是稳定的。T 组询问，$T, n, m \le 10^6$.

- 考虑先放好 m 个位置，这些位置让它就等于 i，然后接下来都必须是不稳定的。
- 发现不稳定的方案等价于 $n-m$ 的错排方案数，设其为 f_{n-m}。
- 考虑递推算 f_n，假设当前已经放好了 $n-1$，要放 n。

SDOI2016 排列计数

题意：求有多少种 n 的排列 A，满足：若第 i 个数 A_i 的值为 i，则称 i 是稳定的。序列恰好有 m 个数是稳定的。T 组询问，$T, n, m \leq 10^6$.

- 考虑先放好 m 个位置，这些位置让它就等于 i，然后接下来都必须是不稳定的。
- 发现不稳定的方案等价于 $n-m$ 的错排方案数，设其为 f_{n-m}。
- 考虑递推算 f_n，假设当前已经放好了 $n-1$，要放 n。
- 假设 n 放在了第 k 位，如果 k 这个数字放在了第 n 位，则去掉这两个位置后，剩下来是 $n-2$ 的错排；如果 k 这个数字没有放在第 n 位，这样等价于去掉第 k 位，然后把 k 当做 n，于是就是 $n-1$ 的错排。

SDOI2016 排列计数

题意：求有多少种 n 的排列 A，满足：若第 i 个数 A_i 的值为 i，则称 i 是稳定的。序列恰好有 m 个数是稳定的。T 组询问，$T, n, m \leq 10^6$.

- 考虑先放好 m 个位置，这些位置让它就等于 i，然后接下来都必须是不稳定的。
- 发现不稳定的方案等价于 $n-m$ 的错排方案数，设其为 f_{n-m}。
- 考虑递推算 f_n，假设当前已经放好了 $n-1$，要放 n。
- 假设 n 放在了第 k 位，如果 k 这个数字放在了第 n 位，则去掉这两个位置后，剩下来是 $n-2$ 的错排；如果 k 这个数字没有放在第 n 位，这样等价于去掉第 k 位，然后把 k 当做 n，于是就是 $n-1$ 的错排。
- k 有 $n-1$ 种选取，所以就有 $f_n = (n-1)(f_{n-1} + f_{n-2})$。

SDOI2016 排列计数

题意：求有多少种 n 的排列 A，满足：若第 i 个数 A_i 的值为 i，则称 i 是稳定的。序列恰好有 m 个数是稳定的。T 组询问，$T, n, m \le 10^6$.

- 考虑先放好 m 个位置，这些位置让它就等于 i，然后接下来都必须是不稳定的。
- 发现不稳定的方案等价于 $n-m$ 的错排方案数，设其为 f_{n-m}。
- 考虑递推算 f_n，假设当前已经放好了 $n-1$，要放 n。
- 假设 n 放在了第 k 位，如果 k 这个数字放在了第 n 位，则去掉这两个位置后，剩下来是 $n-2$ 的错排；如果 k 这个数字没有放在第 n 位，这样等价于去掉第 k 位，然后把 k 当做 n，于是就是 $n-1$ 的错排。
- k 有 $n-1$ 种选取，所以就有 $f_n = (n-1)(f_{n-1}+f_{n-2})$。
- $O(n)$ 预处理 f 和 $n!$ 与 $n!$ 的逆元，就可以 $O(1)$ 应对每组询问了。

AGC 001E BBQ Hard

题意：有 n 个餐包，第 i 个餐包有 a_i 片牛肉和 b_i 片青椒，选出两个不同的餐包将它们的牛肉和青椒串成一串，问有多少种可能的方案。$n \leq 2 * 10^5, 1 \leq a_i, b_i \leq 2000.$

AGC 001E BBQ Hard

- 题意等价于给出 a_i, b_i，求 $\sum_{i=1}^{n} \sum_{j=i+1}^{n} \binom{a_i+a_j+b_i+b_j}{a_i+a_j}$.

AGC 001E BBQ Hard

- 题意等价于给出 a_i, b_i，求 $\sum_{i=1}^{n}\sum_{j=i+1}^{n}\binom{a_i+a_j+b_i+b_j}{a_i+a_j}$.
- 注意到这个二项式系数有组合意义，即从 $(-a_i, -b_i)$ 走到 (a_j, b_j) 的方案数。

AGC 001E BBQ Hard

- 题意等价于给出 a_i, b_i，求 $\sum_{i=1}^{n}\sum_{j=i+1}^{n}\binom{a_i+a_j+b_i+b_j}{a_i+a_j}$.
- 注意到这个二项式系数有组合意义，即从 $(-a_i, -b_i)$ 走到 (a_j, b_j) 的方案数。
- 对每个 j，都去递推其他所有位置走到 (a_j, b_j) 的方案数，递推的初始就是所有的 (a_i, b_i) 处有值。

AGC 001E BBQ Hard

- 题意等价于给出 a_i, b_i，求 $\sum_{i=1}^{n}\sum_{j=i+1}^{n}\binom{a_i+a_j+b_i+b_j}{a_i+a_j}$.
- 注意到这个二项式系数有组合意义，即从 $(-a_i, -b_i)$ 走到 (a_j, b_j) 的方案数。
- 对每个 j，都去递推其他所有位置走到 (a_j, b_j) 的方案数，递推的初始就是所有的 (a_i, b_i) 处有值。
- 注意到这个递推是整体的，所以统一对所有 j 进行递推即可，最后去掉自己对自己的贡献，时间复杂度 $O(ab)$。

鸽巢原理

- 如果 $n+1$ 个物体被放进 n 个盒子，那么至少有一个盒子包含两个或者更多的物体。

鸽巢原理

- 如果 $n+1$ 个物体被放进 n 个盒子，那么至少有一个盒子包含两个或者更多的物体。
- 扩展版本：令 $x_1, x_2, \cdots, x_n$ 为正整数，如果有 $\sum_{k=1}^{n}(x_k-1)+1$ 个物体被放进 n 个盒子，那么至少有一个位置 i 满足第 i 个盒子至少有 x_i 个物体。

鸽巢原理

- 如果 $n+1$ 个物体被放进 n 个盒子，那么至少有一个盒子包含两个或者更多的物体。
- 扩展版本：令 $x_1, x_2, \cdots, x_n$ 为正整数，如果有 $\sum_{k=1}^{n}(x_k-1)+1$ 个物体被放进 n 个盒子，那么至少有一个位置 i 满足第 i 个盒子至少有 x_i 个物体。
- 对于 n 个整数 $a_1, a_2, \cdots, a_n$，存在整数 $1 \le l \le r \le n$，使得 $\sum_{i=l}^{r} a_i$ 是 n 的倍数。

鸽巢原理

- 如果 $n+1$ 个物体被放进 n 个盒子，那么至少有一个盒子包含两个或者更多的物体。
- 扩展版本：令 $x_1, x_2, \cdots, x_n$ 为正整数，如果有 $\sum_{k=1}^{n}(x_k-1)+1$ 个物体被放进 n 个盒子，那么至少有一个位置 i 满足第 i 个盒子至少有 x_i 个物体。
- 对于 n 个整数 $a_1, a_2, \cdots, a_n$，存在整数 $1 \le l \le r \le n$，使得 $\sum_{i=l}^{r} a_i$ 是 n 的倍数。
- 从连续 $2n$ 个数中选 $n+1$ 个数，一定存在两个选择的数之差为 1。

鸽巢原理

- 如果 $n+1$ 个物体被放进 n 个盒子，那么至少有一个盒子包含两个或者更多的物体。
- 扩展版本：令 $x_1, x_2, \cdots, x_n$ 为正整数，如果有 $\sum_{k=1}^{n}(x_k-1)+1$ 个物体被放进 n 个盒子，那么至少有一个位置 i 满足第 i 个盒子至少有 x_i 个物体。
- 对于 n 个整数 $a_1, a_2, \cdots, a_n$，存在整数 $1 \le l \le r \le n$，使得 $\sum_{i=l}^{r} a_i$ 是 n 的倍数。
- 从连续 $2n$ 个数中选 $n+1$ 个数，一定存在两个选择的数之差为 1。
- 若将 n 个物体放入 k 个盒子，则至少有一个盒子包含至少 $\lfloor \frac{n}{k} \rfloor$ 个物体。

鸽巢原理

- 如果 $n+1$ 个物体被放进 n 个盒子，那么至少有一个盒子包含两个或者更多的物体。
- 扩展版本：令 $x_1, x_2, \cdots, x_n$ 为正整数，如果有 $\sum_{k=1}^{n}(x_k - 1) + 1$ 个物体被放进 n 个盒子，那么至少有一个位置 i 满足第 i 个盒子至少有 x_i 个物体。
- 对于 n 个整数 $a_1, a_2, \cdots, a_n$，存在整数 $1 \le l \le r \le n$，使得 $\sum_{i=l}^{r} a_i$ 是 n 的倍数。
- 从连续 $2n$ 个数中选 $n+1$ 个数，一定存在两个选择的数之差为 1。
- 若将 n 个物体放入 k 个盒子，则至少有一个盒子包含至少 $\lfloor \frac{n}{k} \rfloor$ 个物体。
- 由 n^2+1 个数组成的排列要么包含长度为 $n+1$ 的严格递增子序列，要么包含长度为 $n+1$ 的严格递减子序列。

鸽巢原理

- 如果 $n+1$ 个物体被放进 n 个盒子，那么至少有一个盒子包含两个或者更多的物体。
- 扩展版本：令 $x_1, x_2, \cdots, x_n$ 为正整数，如果有 $\sum_{k=1}^{n}(x_k-1)+1$ 个物体被放进 n 个盒子，那么至少有一个位置 i 满足第 i 个盒子至少有 x_i 个物体。
- 对于 n 个整数 $a_1, a_2, \cdots, a_n$，存在整数 $1 \le l \le r \le n$，使得 $\sum_{i=l}^{r} a_i$ 是 n 的倍数。
- 从连续 $2n$ 个数中选 $n+1$ 个数，一定存在两个选择的数之差为 1。
- 若将 n 个物体放入 k 个盒子，则至少有一个盒子包含至少 $\lfloor \frac{n}{k} \rfloor$ 个物体。
- 由 n^2+1 个数组成的排列要么包含长度为 $n+1$ 的严格递增子序列，要么包含长度为 $n+1$ 的严格递减子序列。
- Ramsey 定理，感兴趣可以自行了解。

容斥原理

容斥原理算是既简单又非常难的原理了，经常出现容斥的题，简单的很简单，难的却很难，相当重要。

- 对于集合 S 的 m 个子集 $S_1, S_2, \cdots, S_m$，有 $|\cup_{i=1}^{m} S_i| = \sum_{k=1}^{m}(-1)^{k-1}\sum_{1\le i_1<i_2<\cdots<i_k\le m}|S_{i_1}\cap S_{i_2}\cap\cdots S_{i_k}|$.

容斥原理

容斥原理算是既简单又非常难的原理了，经常出现容斥的题，简单的很简单，难的却很难，相当重要。

- 对于集合 S 的 m 个子集 $S_1, S_2, \cdots, S_m$，有 $|\cup_{i=1}^m S_i| = \sum_{k=1}^m (-1)^{k-1} \sum_{1\le i_1<i_2<\cdots<i_k\le m} |S_{i_1} \cap S_{i_2} \cap \cdots S_{i_k}|$.
- $\min\max$ 容斥：对于集合 $S = \{a_1, \cdots, a_m\}$，有 $\max\{S\} = \sum_{k=1}^m (-1)^{k-1} \sum_{1\le i_1<i_2<\cdots<i_k\le m} \min\{a_{i_1}, \cdots, a_{i_k}\}$.
- 交换 $\min$ 和 $\max$ 也是正确的，加上期望也是正确的，证明写写 $\sum$ 就能解决。

51nod 1667 概率好题

题意：给定 n 个整数区间 $[L_i, R_i]$，随机在第 i 个区间里选出一个整数 x_i，令 $T = \sum_{i=1}^{n} x_i$，求 $T < 0$ 的概率。
$1 \le n \le 16, -10^7 \le L_i \le R_i \le 10^7$.

51nod 1667 概率好题

题意：给定 n 个整数区间 $[L_i, R_i]$，随机在第 i 个区间里选出一个整数 x_i，令 $T = \sum_{i=1}^{n} x_i$，求 $T < 0$ 的概率。
$1 \le n \le 16, -10^7 \le L_i \le R_i \le 10^7$.

- < 0 其实不太容易处理，考虑再加一个变量 $x_{n+1} \ge 1$，求 $\sum_{i=1}^{n+1} x_i = 0$.

51nod 1667 概率好题

题意：给定 n 个整数区间 $[L_i, R_i]$，随机在第 i 个区间里选出一个整数 x_i，令 $T = \sum_{i=1}^{n} x_i$，求 $T < 0$ 的概率。
$1 \le n \le 16, -10^7 \le L_i \le R_i \le 10^7$.

- < 0 其实不太容易处理，考虑再加一个变量 $x_{n+1} \ge 1$，求 $\sum_{i=1}^{n+1} x_i = 0$.
- 这是带上下界的隔板问题，下界容易处理，统一减掉自己的下界就行了，于是题意转化成一堆在 $[0, R_i]$ 的变量，要求和为 T.

51nod 1667 概率好题

题意：给定 n 个整数区间 $[L_i, R_i]$，随机在第 i 个区间里选出一个整数 x_i，令 $T=\sum_{i=1}^{n} x_i$，求 $T<0$ 的概率。
$1 \le n \le 16, -10^7 \le L_i \le R_i \le 10^7$.

- <0 其实不太容易处理，考虑再加一个变量 $x_{n+1} \ge 1$，求 $\sum_{i=1}^{n+1} x_i = 0$.
- 这是带上下界的隔板问题，下界容易处理，统一减掉自己的下界就行了，于是题意转化成一堆在 $[0, R_i]$ 的变量，要求和为 T.
- 发现直接其实并不好算，考虑容斥原理，设 S 表示 x_i 无限制但 $\sum_{i=1}^{n+1} x_i = T$ 的 $\{x_1, \cdots, x_{n+1}\}$ 集合，子集 S_i 分别表示 $0 \le x_i \le R_i$ 且 $\sum_{i=1}^{n+1} x_i = T$ 的 $\{x_1, \cdots, x_{n+1}\}$ 集合，发现 S_i 的交集确实是答案，但我们的容斥原理是并集，这该怎么办呢？

51nod 1667 概率好题

- 发现直接其实并不好算，考虑容斥原理，设 S 表示 x_i 无限制但 $\sum_{i=1}^{n+1} x_i = T$ 的 $\{x_1, \cdots, x_{n+1}\}$ 集合，子集 S_i 分别表示 $0 \leq x_i \leq R_i$ 且 $\sum_{i=1}^{n+1} x_i = T$ 的 $\{x_1, \cdots, x_{n+1}\}$ 集合，发现 S_i 的交集确实是答案，但我们的容斥原理是并集，这该怎么办呢?

51nod 1667 概率好题

- 发现直接其实并不好算，考虑容斥原理，设 S 表示 x_i 无限制但 $\sum_{i=1}^{n+1} x_i = T$ 的 $\{x_1, \cdots, x_{n+1}\}$ 集合，子集 S_i 分别表示 $0 \le x_i \le R_i$ 且 $\sum_{i=1}^{n+1} x_i = T$ 的 $\{x_1, \cdots, x_{n+1}\}$ 集合，发现 S_i 的交集确实是答案，但我们的容斥原理是并集，这该怎么办呢?
- 集合的交并有一个很简单的性质，$|\cap_{i=1}^{n} S_i| = |S| - |\cup_{i=1}^{n} \overline{S_i}|$，而 $|S|$ 很好求，这是最直接的隔板法，剩下来我们想想 $\cap \overline{S_i}$ 咋求。

51nod 1667 概率好题

- 发现直接其实并不好算，考虑容斥原理，设 S 表示 x_i 无限制但 $\sum_{i=1}^{n+1} x_i = T$ 的 $\{x_1, \cdots, x_{n+1}\}$ 集合，子集 S_i 分别表示 $0 \le x_i \le R_i$ 且 $\sum_{i=1}^{n+1} x_i = T$ 的 $\{x_1, \cdots, x_{n+1}\}$ 集合，发现 S_i 的交集确实是答案，但我们的容斥原理是并集，这该怎么办呢?
- 集合的交并有一个很简单的性质，$|\cap_{i=1}^{n} S_i| = |S| - |\cup_{i=1}^{n} \overline{S_i}|$，而 $|S|$ 很好求，这是最直接的隔板法，剩下来我们想想 $\cap \overline{S_i}$ 咋求。
- $\cap \overline{S_i}$ 要求的是某一些 x_i 必须超过 R_i，这又是经典的带下界隔板法，直接等式两边同时减去 R_i，就变成最基础的隔板法了。时间复杂度 $O(2^n)$.

HDU 6397 Character Encoding

题意：给定 n, m, k，求整数方程 $\sum_{i=1}^{m} x_i = k, 0 \le x_i < n$ 的解数。$n, m, k \le 10^6$.

HDU 6397 Character Encoding

题意：给定 n, m, k，求整数方程 $\sum_{i=1}^{m} x_i = k, 0 \le x_i < n$ 的解数。$n, m, k \le 10^6$.

- 和上一题一致，直接考虑强迫 i 个数超过限制，然后进行容斥。

HDU 6397 Character Encoding

题意：给定 n, m, k，求整数方程 $\sum_{i=1}^{m} x_i = k, 0 \le x_i < n$ 的解数。$n, m, k \le 10^6$.

- 和上一题一致，直接考虑强迫 i 个数超过限制，然后进行容斥。
- 强迫 i 个数超过限制，这样的挑选方案有 $\binom{m}{i}$ 种，然后隔板法得到另一部分是 $\binom{k-i*n+m-1}{m-1}$，这里的二项式系数仍然可以预处理阶乘和阶乘的逆元得到，总时间复杂度 $O(n+m+k)$。

The 2021 CCPC Weihai Onsite M 810975

题意：给一个 n, m, k，构造一个 n 位的 01 串，其中有 m 个 1 并且最长的连续的 1 长度为 k。$n, m, k \leq 10^5$.

The 2021 CCPC Weihai Onsite M 810975

题意：给一个 n, m, k，构造一个 n 位的 01 串，其中有 m 个 1 并且最长的连续的 1 长度为 k。$n, m, k \leq 10^5$.

- 考虑理解成 n 个格子，插入 $n-m$ 个 0，使得 0 之间的最大间隔为 k.

The 2021 CCPC Weihai Onsite M 810975

题意：给一个 n, m, k，构造一个 n 位的 01 串，其中有 m 个 1 并且最长的连续的 1 长度为 k。$n, m, k \le 10^5$.

- 考虑理解成 n 个格子，插入 $n - m$ 个 0，使得 0 之间的最大间隔为 k.
- 容易发现等于 k 并不让人舒服，不过可以看成 $\le k$ 减去 $\le k - 1$。

The 2021 CCPC Weihai Onsite M 810975

题意：给一个 n, m, k，构造一个 n 位的 01 串，其中有 m 个 1 并且最长的连续的 1 长度为 k。$n, m, k \le 10^5$.

- 考虑理解成 n 个格子，插入 $n-m$ 个 0，使得 0 之间的最大间隔为 k.
- 容易发现等于 k 并不让人舒服，不过可以看成 $\le k$ 减去 $\le k-1$。
- 于是就等价于，有 $n-m+1$ 个数 x_i，满足 $0 \le x_i \le k$，且 $\sum_{i=1}^{n-m+1} x_i = m$ 的方案数。

The 2021 CCPC Weihai Onsite M 810975

题意：给一个 n, m, k，构造一个 n 位的 01 串，其中有 m 个 1 并且最长的连续的 1 长度为 k。$n, m, k \le 10^5$.

- 考虑理解成 n 个格子，插入 $n-m$ 个 0，使得 0 之间的最大间隔为 k.
- 容易发现等于 k 并不让人舒服，不过可以看成 $\le k$ 减去 $\le k-1$。
- 于是就等价于，有 $n-m+1$ 个数 x_i，满足 $0 \le x_i \le k$，且 $\sum_{i=1}^{n-m+1} x_i = m$ 的方案数。
- 带上界的隔板法，直接容斥即可。时间复杂度 $O(n+m+k)$.

CF 559C Gerald and Giant Chess

题意：平面上有 k 个障碍，问从 $(1,1)$ 出发，每走一步要么横坐标加一，要么纵坐标加一，走到 (n,m) 不经过任何一个障碍的路径数。$n,m \leq 10^5, k \leq 2000.$

CF 559C Gerald and Giant Chess

题意：平面上有 k 个障碍，问从 $(1,1)$ 出发，每走一步要么横坐标加一，要么纵坐标加一，走到 (n,m) 不经过任何一个障碍的路径数。$n,m \le 10^5, k \le 2000.$

- 仍然将 S 看成所有走到 (n,m) 的路径，子集 S_i 看成不经过第 i 个点到达 (n,m) 的路径，这样答案就是 S_i 的交集。同样地，将交集转成并集，于是就是求 S_i 的补集的交。

CF 559C Gerald and Giant Chess

题意：平面上有 k 个障碍，问从 $(1,1)$ 出发，每走一步要么横坐标加一，要么纵坐标加一，走到 (n,m) 不经过任何一个障碍的路径数。$n,m \le 10^5, k \le 2000$.

- 仍然将 S 看成所有走到 (n,m) 的路径，子集 S_i 看成不经过第 i 个点到达 (n,m) 的路径，这样答案就是 S_i 的交集。同样地，将交集转成并集，于是就是求 S_i 的补集的交。
- 直接暴力枚举就 $O(2^k)$ 了，有没有更好的想法呢?

CF 559C Gerald and Giant Chess

题意：平面上有 k 个障碍，问从 $(1,1)$ 出发，每走一步要么横坐标加一，要么纵坐标加一，走到 (n,m) 不经过任何一个障碍的路径数。$n, m \leq 10^5, k \leq 2000.$

- 仍然将 S 看成所有走到 (n,m) 的路径，子集 S_i 看成不经过第 i 个点到达 (n,m) 的路径，这样答案就是 S_i 的交集。同样地，将交集转成并集，于是就是求 S_i 的补集的交。
- 直接暴力枚举就 $O(2^k)$ 了，有没有更好的想法呢?
- 其实我们可以将容斥系数直接蕴于动态规划，这又是什么意思呢?

CF 559C Gerald and Giant Chess

题意：平面上有 k 个障碍，问从 $(1, 1)$ 出发，每走一步要么横坐标加一，要么纵坐标加一，走到 (n, m) 不经过任何一个障碍的路径数。$n, m \le 10^5, k \le 2000.$

- 仍然将 S 看成所有走到 (n, m) 的路径，子集 S_i 看成不经过第 i 个点到达 (n, m) 的路径，这样答案就是 S_i 的交集。同样地，将交集转成并集，于是就是求 S_i 的补集的交。
- 直接暴力枚举就 $O(2^k)$ 了，有没有更好的想法呢?
- 其实我们可以将容斥系数直接蕴于动态规划，这又是什么意思呢?
- 将格子按可达顺序排序后 (按横纵坐标递增)，设 f_i 表示从起点到第 i 个不可走的格子，途中不经过任何不可走的格子的方案数，这个东西的值等价于前 i 个 S_j 的交，对这个东西进行容斥计算。

CF 559C Gerald and Giant Chess

- 设 f_i 表示从起点到第 i 个不可走的格子，途中不经过任何不可走的格子的方案数，对这个东西进行容斥计算。

CF 559C Gerald and Giant Chess

- 设 f_i 表示从起点到第 i 个不可走的格子，途中不经过任何不可走的格子的方案数，对这个东西进行容斥计算。
- 这个东西等价于 $|\cap_{j=1}^{i-1} S_{i,j}|$，同样的交转并，于是要求的是 $|S_i|-\sum_{k=1}^{i-1}(-1)^{k-1}\sum_{1\le i_1<i_2<\cdots<i_k\le i-1}|\overline{S_{i,i_1}}\cap\overline{S_{i,i_2}}\cap\cdots\overline{S_{i,i_k}}|$. 注意到此时的全集 S_i 指的是到达第 i 个格子的方案。

CF 559C Gerald and Giant Chess

- 设 f_i 表示从起点到第 i 个不可走的格子，途中不经过任何不可走的格子的方案数，对这个东西进行容斥计算。
- 这个东西等价于 $|\cap_{j=1}^{i-1} S_{i,j}|$，同样的交转并，于是要求的是 $|S_i|-\sum_{k=1}^{i-1}(-1)^{k-1}\sum_{1\le i_1<i_2<\cdots<i_k\le i-1}|\overline{S_{i,i_1}}\cap\overline{S_{i,i_2}}\cap\cdots\overline{S_{i,i_k}}|$. 注意到此时的全集 S_i 指的是到达第 i 个格子的方案。
- 有趣的是，我们发现 $|\overline{S_{j,i_1}}\cap\overline{S_{j,i_2}}\cap\cdots\overline{S_{j,i_k}}|$ 乘上 $P(i,j)$ 的时候恰好等于 $|\overline{S_{i,i_1}}\cap\overline{S_{i,i_2}}\cap\cdots\overline{S_{i,i_k}}\cap\overline{S_{i,j}}|$，其中 $P(i,j)$ 表示从 j 到 i 的方案数，这个方案数是直接组合数表示的，即 $\binom{x_i+y_i-x_j-y_j}{x_i-x_j}$. 这个证明直接考虑这个组合意义即可，即所有先到达 i_1 然后到 j 再到 i 的路径恰好等价于先到达 i_1 再到 i 的路径与先到 j 再到 i 的路径的交。

CF 559C Gerald and Giant Chess

- 设 f_i 表示从起点到第 i 个不可走的格子，途中不经过任何不可走的格子的方案数，对这个东西进行容斥计算。
- 这个东西等价于 $|\cap_{j=1}^{i-1} S_{i,j}|$，同样的交转并，于是要求的是 $|S_i|-\sum_{k=1}^{i-1}(-1)^{k-1}\sum_{1\le i_1<i_2<\cdots<i_k\le i-1}|\overline{S_{i,i_1}}\cap\overline{S_{i,i_2}}\cap\cdots\overline{S_{i,i_k}}|$. 注意到此时的全集 S_i 指的是到达第 i 个格子的方案。
- 有趣的是，我们发现 $|\overline{S_{j,i_1}}\cap\overline{S_{j,i_2}}\cap\cdots\overline{S_{j,i_k}}|$ 乘上 $P(i,j)$ 的时候恰好等于 $|\overline{S_{i,i_1}}\cap\overline{S_{i,i_2}}\cap\cdots\overline{S_{i,i_k}}\cap\overline{S_{i,j}}|$，其中 $P(i,j)$ 表示从 j 到 i 的方案数，这个方案数是直接组合数表示的，即 $\binom{x_i+y_i-x_j-y_j}{x_i-x_j}$. 这个证明直接考虑这个组合意义即可，即所有先到达 i_1 然后到 j 再到 i 的路径恰好等价于先到达 i_1 再到 i 的路径与先到 j 再到 i 的路径的交。
- 好像发现了什么？疑似 $f_i=|S_i|-\sum_{j=0}^{i-1}f_j\times\binom{x_i+y_i-x_j-y_j}{x_i-x_j}$.

CF 559C Gerald and Giant Chess

- 好像发现了什么？疑似 $f_i = |S_i| - \sum_{j=0}^{i-1} f_j \times \binom{x_i+y_i-x_j-y_j}{x_i-x_j}$.

CF 559C Gerald and Giant Chess

- 好像发现了什么？疑似 $f_i = |S_i| - \sum_{j=0}^{i-1} f_j \times \binom{x_i+y_i-x_j-y_j}{x_i-x_j}$.
- 严格证明就利用当前的等式证明，直观感受也很合理，因为恰好每一项交多了一项，然后 (-1) 的幂次转一下，恰好符号很是正确。另一方面，从组合意义来看，先是所有到 i 的路径，然后每次去掉从其他点走过来的路径，也很合理。

CF 559C Gerald and Giant Chess

- 好像发现了什么？疑似 $f_i = |S_i| - \sum_{j=0}^{i-1} f_j \times \binom{x_i+y_i-x_j-y_j}{x_i-x_j}$.
- 严格证明就利用当前的等式证明，直观感受也很合理，因为恰好每一项交多了一项，然后 (-1) 的幂次转一下，恰好符号很是正确。另一方面，从组合意义来看，先是所有到 i 的路径，然后每次去掉从其他点走过来的路径，也很合理。
- 于是预处理组合数就可以 $O(k^2)$ 求出此题了，当然如果多项式水平比较不错的可能会发现更优做法。

CF 559C Gerald and Giant Chess

- 好像发现了什么？疑似 $f_i = |S_i| - \sum_{j=0}^{i-1} f_j \times \binom{x_i+y_i-x_j-y_j}{x_i-x_j}$.
- 严格证明就利用当前的等式证明，直观感受也很合理，因为恰好每一项交多了一项，然后 (-1) 的幂次转一下，恰好符号很是正确。另一方面，从组合意义来看，先是所有到 i 的路径，然后每次去掉从其他点走过来的路径，也很合理。
- 于是预处理组合数就可以 $O(k^2)$ 求出此题了，当然如果多项式水平比较不错的可能会发现更优做法。
- 平时做题的时候，这种直观感受就已经足够了，并没有必要都严格证明容斥的正确性。

CF 559C Gerald and Giant Chess

- 还有一个角度去理解这样的容斥。我们先考虑一个大暴力，设 $f_{i,S}$ 表示当前算到了第 i 个点，枚举的子集是 S 的方案数。

CF 559C Gerald and Giant Chess

- 还有一个角度去理解这样的容斥。我们先考虑一个大暴力，设 $f_{i,S}$ 表示当前算到了第 i 个点，枚举的子集是 S 的方案数。
- 我们发现，转移时，S 的作用仅仅是决定枚举的子集是奇数个元素还是偶数个元素，也即前面的系数是 1 还是 -1。于是我们可以把前面系数相同的子集一起计算。

CF 559C Gerald and Giant Chess

- 还有一个角度去理解这样的容斥。我们先考虑一个大暴力，设 $f_{i,S}$ 表示当前算到了第 i 个点，枚举的子集是 S 的方案数。
- 我们发现，转移时，S 的作用仅仅是决定枚举的子集是奇数个元素还是偶数个元素，也即前面的系数是 1 还是 -1。于是我们可以把前面系数相同的子集一起计算。
- 设 $f_{i,0/1}$ 表示当前算到了第 i 个点，系数是 1 或者 -1 的方案数，容易发现转移式和之前的做法是相同的。于是我们就从另一个角度得到了相同的做法了。

ZJOI2016 小星星

题意：给出一张图 G 和一棵树 T，点数都为 n，求有多少个长为 n 的排列 a_i，满足若存在 $(u,v) \in T$ 则一定存在 $(a_u, a_v) \in G$。
$n \leq 17, m \leq \frac{n(n-1)}{2}$.

ZJOI2016 小星星

题意：给出一张图 G 和一棵树 T，点数都为 n，求有多少个长为 n 的排列 a_i，满足若存在 $(u,v)\in T$ 则一定存在 $(a_u,a_v)\in G$。$n\le 17, m\le \frac{n(n-1)}{2}$.

- 考虑一个暴力的状压 dp，设 $f_{i,j,S}$ 表示 i 节点映射成 j，已经用过的编号集合是 S 的方案数。这样直接按着树来深度优先搜索，自下向上合并子树即可，但合并时要枚举 S 的子集，所以时间复杂度是 $O(n^3 3^n)$. 无法通过。

ZJOI2016 小星星

题意：给出一张图 G 和一棵树 T，点数都为 n，求有多少个长为 n 的排列 a_i，满足若存在 $(u,v)\in T$ 则一定存在 $(a_u,a_v)\in G$。$n\le 17, m\le \frac{n(n-1)}{2}$.

- 考虑一个暴力的状压 dp，设 $f_{i,j,S}$ 表示 i 节点映射成 j，已经用过的编号集合是 S 的方案数。这样直接按着树来深度优先搜索，自下向上合并子树即可，但合并时要枚举 S 的子集，所以时间复杂度是 $O(n^3 3^n)$. 无法通过。
- 发现瓶颈是枚举子集，枚举子集是为了恰好是 n 的排列这个条件，更进一步，可以理解成 1 到 n 每个编号都用过一遍。能否放松这个条件呢？可以，只用过一遍这个条件可以容斥掉。

ZJOI2016 小星星

题意：给出一张图 G 和一棵树 T，点数都为 n，求有多少个长为 n 的排列 a_i，满足若存在 $(u,v)\in T$ 则一定存在 $(a_u,a_v)\in G$。
$n\leq 17, m\leq \frac{n(n-1)}{2}$.

- 考虑一个暴力的状压 dp，设 $f_{i,j,S}$ 表示 i 节点映射成 j，已经用过的编号集合是 S 的方案数。这样直接按着树来深度优先搜索，自下向上合并子树即可，但合并时要枚举 S 的子集，所以时间复杂度是 $O(n^3 3^n)$. 无法通过。
- 发现瓶颈是枚举子集，枚举子集是为了恰好是 n 的排列这个条件，更进一步，可以理解成 1 到 n 每个编号都用过一遍。能否放松这个条件呢？可以，只用过一遍这个条件可以容斥掉。
- 这里的容斥中，S 显然表示所有合法映射，S_i 表示用了 i 这个编号至少一次的合法映射，发现 $\cap S_i$ 正是答案。照常地交转并，剩下来的就是求补集的交。

ZJOI2016 小星星

- S_i 的补集是 i 这个编号一次都不能用过，那么交集就是一堆编号不能使用，然后其他编号任意使用。这是可以轻松动态规划的，设 $f_{i,j}$ 表示 i 节点映射成 j 的方案数，直接深度优先搜索一遍树就可以算出答案了。于是枚举交集，再深搜合并子树就完成了这道题，时间复杂度 $O(n^3 2^n)$.

ZJOI2016 小星星

- S_i 的补集是 i 这个编号一次都不能用过，那么交集就是一堆编号不能使用，然后其他编号任意使用。这是可以轻松动态规划的，设 $f_{i,j}$ 表示 i 节点映射成 j 的方案数，直接深度优先搜索一遍树就可以算出答案了。于是枚举交集，再深搜合并子树就完成了这道题，时间复杂度 $O(n^3 2^n)$.
- 这题的容斥也有直观感受的，可以理解成至多用了 n 个编号减至多用了 $n-1$ 个编号加至多用了 $n-2$ 个的以此类推。

HDU 4366 Card Collector

题意：小浣熊干脆面里装有英雄卡，有 n 种英雄卡，一包干脆面里有 p_i 的概率出现第 i 张英雄卡，想知道在期望意义上买多少包干脆面可以把英雄卡集齐。$n \leq 20$.

HDU 4366 Card Collector

题意：小浣熊干脆面里装有英雄卡，有 n 种英雄卡，一包干脆面里有 p_i 的概率出现第 i 张英雄卡，想知道在期望意义上买多少包干脆面可以把英雄卡集齐。$n \leq 20$.

- 可以发现，全部出现这个条件非常苛刻，但出现一张却很容易，这启发我们考虑 $\min\max$ 容斥。

HDU 4366 Card Collector

题意：小浣熊干脆面里装有英雄卡，有 n 种英雄卡，一包干脆面里有 p_i 的概率出现第 i 张英雄卡，想知道在期望意义上买多少包干脆面可以把英雄卡集齐。$n \le 20$.

- 可以发现，全部出现这个条件非常苛刻，但出现一张却很容易，这启发我们考虑 $\min\max$ 容斥。
- $\min\max$ 容斥的式子再写一遍，是 $E(\max_{i\in S} x_i) = \sum_{T\subseteq S}(-1)^{|T|-1}E(\min_{j\in T} x_j)$. 此时期望的意义恰好是我们想要的，我们令 x_i 表示元素 i 出现的时间。那么 $\max$ 指的是最迟出现的元素的时间，$\min$ 是最早的。再进一步，$\max$ 是整个集合出现的时间，$\min$ 就是集合中最早出现的元素的时间。我们要求的是 $E(S)$，此时 S 为全集。

HDU 4366 Card Collector

题意：小浣熊干脆面里装有英雄卡，有 n 种英雄卡，一包干脆面里有 p_i 的概率出现第 i 张英雄卡，想知道在期望意义上买多少包干脆面可以把英雄卡集齐。$n \leq 20$.

- 可以发现，全部出现这个条件非常苛刻，但出现一张却很容易，这启发我们考虑 $\min\max$ 容斥。
- $\min\max$ 容斥的式子再写一遍，是 $E(\max_{i\in S} x_i) = \sum_{T\subseteq S}(-1)^{|T|-1}E(\min_{j\in T} x_j)$. 此时期望的意义恰好是我们想要的，我们令 x_i 表示元素 i 出现的时间。那么 $\max$ 指的是最迟出现的元素的时间，$\min$ 是最早的。再进一步，$\max$ 是整个集合出现的时间，$\min$ 就是集合中最早出现的元素的时间。我们要求的是 $E(S)$，此时 S 为全集。
- 发现 $E(\min_{i\in S} x_i)$ 很好求，考虑 $P = \sum_{i\in S} p_i$，于是期望时间就是经典的等比数列，$\sum_{i\geq 0}(i+1)P(1-P)^i = \frac{1}{P}$，所以直接暴力递归就求出答案了，时间复杂度 $O(2^n)$.

一些习题

PE 364 Comfortable distance

AGC 002 F

CERC2015 Frightful Formula

CSP-S2019 Emiya 家今天的饭

一些习题

SDWC2018 Day1 网格

PE 677 Coloured Graphs

ZJOI2022 树

LibreOJ NOI Round 2 不等关系

2016 Petrozavodsk Winter, Makoto Soejima Contest 4 H Random Walk

2016 Petrozavodsk Winter, Makoto Soejima Contest 4 G Paint

XVI Open Cup, Grand Prix of Ukraine J Joining Powers

一些习题

PKUWC2018 随机游走

集训队作业 2018 小 Z 的礼物

LG 4707 重返现世

目录

总结

本来还想着讲讲组合反演以及各种特殊数的，后来想了下，核心要点应该是要讲容斥，最后就放弃拓展更多内容了。

那么，有缘再会了。祝大家身体健康！