概率与期望

daklqw

Zhenhai High School

August 4, 2023

前言

OI 中概率论部分一般只要求掌握基本的定义和性质。
主要考察动态规划以及数据结构等知识的功底。

这次讲课会带点形式化的内容，可能需要点微积分和集合论的知识来辅助理解。
但是是属于拓展内容，相信大家打 OI 应该用不到，所以大家也不必产生符号恐惧。
不过我觉得对深入的理解还是有帮助的，毕竟省选往上，OI 有不少是数学。
然后水平有限，写出的内容可能不严谨，请多包涵。

样本空间与事件

假设有一个试验，重复这个试验，会产生不同的结果且不能确定这个结果，那么称这个试验为随机试验。

随机试验所有可能产生的、不能再细分的结果，称为基本结果。

由所有基本结果组成的集合，称为样本空间。

样本空间的任一子集称为事件。

由所有基本结果组成的集合，称为样本空间。
例如，有一个盒子，里面一共有六个球：

- 编号为 $1, 2, 3$ 的 红球，
- 编号为 $4, 5, 6$ 的 蓝球，

随机地从盒子里取一个球，一共有六种结果。
那么样本空间是 $\{1, 2, 3, 4, 5, 6\}$。
样本空间的任一子集称为事件。
子集 $\{1, 2, 3\}$ 对应事件“取出了一个红球”。
子集 $\{1, 3, 5\}$ 对应事件“取出了一个编号为奇数的球”。

事件的形式化定义

Definition (事件)

设 $\mathcal{F}$ 是一个集合 Ω 的一些子集组成的集合，满足

1. $\Omega \in \mathcal{F}$.
2. $\forall A \in \mathcal{F}$，那么 $\overline{A} := \Omega \setminus A \in \mathcal{F}$.
3. 对任意列 $A_i \in \mathcal{F}$，那么 $\bigcup\limits_{i=1}^{+\infty} A_i \in \mathcal{F}$.

那么称 $\mathcal{F}$ 为事件体，称 $A \in \mathcal{F}$ 为事件，称 $\overline{A}$ 为 A 的逆事件。
特别地，称 Ω 为必然事件，$\emptyset$ 为不可能事件。

Ω 可以看作是样本空间。

1. $\Omega \in \mathcal{F}$.
2. $\forall A \in \mathcal{F}$，那么 $\overline{A} := \Omega \setminus A \in \mathcal{F}$.
3. 对任意列 $A_i \in \mathcal{F}$，那么 $\bigcup_{i=1}^{+\infty} A_i \in \mathcal{F}$.

这三条说明，事件对于可数无穷多次以内的交、并、补运算是封闭的。

对于两个事件 A, B，可以用 $AB = A \cap B$ 表示两个事件的交。

概率

概率是用一个实数来衡量一个事件发生的可能性大小。事件的概率值可以看成事件为自变量的函数值。

例如，盒子里有六个球：$\{1, 2, 3, 4, 5, 6\}$，如果取到每个球的可能性是一样的，那么去一个球是红色的概率便是 $50\% = 0.5$。

概率的大小被限制在 $[0, 1]$。

概率

Definition (概率)

概率 $P:\mathcal{F}\to\mathbb{R}$ 是满足下面三个条件的函数：

1. $\forall A\in\mathcal{F}, 0\le P(A)\le 1.$
2. $P(\Omega)=1.$
3. 对于任意两两不交的列 $A_i\in\mathcal{F}$，有
$$P\left(\bigcup_{i=1}^{+\infty}A_i\right)=\sum_{i=1}^{+\infty}P(A_i).$$

三元组 $(\Omega,\mathcal{F},P)$ 称为概率空间。

- 对于任意两两不交的列 $A_i \in \mathcal{F}$，有

$$P\left(\bigcup_{i=1}^{+\infty} A_i\right) = \sum_{i=1}^{+\infty} P(A_i).$$

注意到第三条要求两两不交。其实对于有交的 A_i 来计算 $P(\cup A_i)$ 可以通过容斥原理（这里要求 A_i 是有限的）：

$$P\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{S\subseteq\{1,\dots,n\},S\neq\emptyset} (-1)^{|S|} P\left(\bigcap_{i\in S} A_i\right).$$

古典概率模型

如果一个试验只有有限个基本结果，且每个基本结果出现的可能性是一样的，那么这样的模型称为古典概率模型。

此时一个事件 A 发生的概率就是 $P(A)=\dfrac{|A|}{|\Omega|}$。

例如在一个有着有限球的盒子里等概率随机拿一个球，这样的试验模型就是古典概率模型。

几何概率模型

当 Ω 是空间中的一个区域，对于事件 A，有 $P(A)=\dfrac{\mu(A)}{\mu(\Omega)}$，其中 μ 是其测度（例如面积、体积等），那么将这个模型称为几何概率模型。

和古典概型相比，其 Ω 是不可列的。但是仍然是均匀分布的。

Definition (事件的独立性)

称事件 A 和事件 B 相互独立，如果

$$P(AB) = P(A)P(B)$$

易知任意事件 A 和 $\emptyset, \Omega$ 独立。

Property

若 A 和 B 相互独立，那么 $\overline{A}$ 和 $\overline{B}$ 相互独立。

OI 中的概率计算

虽然说概率是个实数，但是实数在某些状态数巨大问题中很容易产生误差，所以部分题目会采用取模的方式，例如在模素数 P 域下进行计算。

一般来说，OI 中出现的模型是古典概率模型。此时概率可以表示成有理数之比 $\frac{a}{b}$ 的形式。

模素数 P 的情况下，如果 b 非零，则一定存在逆元 b^{-1}，则该概率在模 P 意义下就是 ab^{-1}。

简单给一下逆元计算方法

一种是通过费马小定理，使用快速幂计算 a^{P-2} 得到。
一种是使用 exgcd。
还有一种是通过递推的方式，有 $i^{-1} = -\left\lfloor \frac{P}{i} \right\rfloor (p \mod i)^{-1}$。
如果不会写，可以去参考 OI-Wiki 的乘法逆元部分。

Example (考场奇遇)

Alice 和 Bob 去参加一场考试，一共有 n 道判断题。
Alice 考完后和 Bob 对了答案，知道了自己有哪些题目作答和他相同，哪些不同。
现在已知 Bob 每道题的作答正确的概率是 $p\%$，求 Alice 作答正确至少 m 道题的概率。
要求 $O(n^2)$ 时间复杂度。

Bob 每道判断题正确概率是 $p\%$，求 Alice 至少对 m 道题的概率。

对于作答相同的题目，Alice 正确当且仅当 Bob 正确。

对于作答不同的题目，Alice 正确当且仅当 Bob 错误。

所以这里的基本事件，是 Bob 在第 i 道题上作答正确/错误，对应事件 T_i/F_i。目前已知 $P(T_i) = p, P(F_i) = 1 - p$。

不同题目作答是否正确是相互独立的，同一道题目的 T_i 和 F_i 也是不交的。

所以很容易计算出一种作答状态对应的概率。例如 $P(T_1F_2F_3) = p(1-p)^2$。

不同题目作答是否正确是相互独立的，同一道题目的 T_i 和 F_i 也是不交的。

这说明可以枚举满足条件（Alice 正确至少 m 道）的作答情况，来计算出答案。

由独立性，可以通过 DP 计算前 i 道题，Alice 对 j 道的概率 $f_{i,j}$。

并且枚举第 $i+1$ 道题对应的事件，由独立性追加到 $f_{i,j}$ 对应的事件上，转移到 $f_{i+1,j}$ 和 $f_{i+1,j+1}$ 上。

这样就做到了 $O(n^2)$ 时间复杂度。

令 k 为两人相同的题数，那么 $n-k$ 为不同的题数。其实可以直接使用组合数枚举相同的题中有哪些正确，不同的题中有哪些正确：

$$Ans=\sum_{a=0}^{k}\sum_{b=0}^{n-k}[a+b\geq m]\binom{k}{a}\binom{n-k}{b}p^{a+n-k-b}(1-p)^{b+k-a}.$$

通过预处理组合数同样可以做到 $O(n^2)$ 时间复杂度。

由独立性，可以通过 DP 计算前 i 道题，Alice 对 j 道的概率 $f_{i,j}$。
实际上这 DP 对应着生成函数

$$(1-p+px)^a(p+(1-p)x)^b.$$

通过类似于背包的计算方法同样可以做到 $O(n^2)$ 时间复杂度。

Example (「PKUWC2018」随机算法)

给定一张无向简单图，用以下算法尝试求图 $G=(V,E)$ 的最大独立集 S：

1. 等概率随机一个 $1\dots n=|V|$ 的排列 p。
2. 维护答案集合 S，一开始 $S=\emptyset$。
3. 按照 $i=1\dots n$ 的顺序，检查 $S\cup\{p_i\}$ 是否是一个独立集，如果是就令 $S=S\cup\{p_i\}$。

求最后的 S 得到原图最大独立集的概率。

数据范围：$1\le n\le 20$。

讲个故事，当年整场比赛全是计数题。

只要求出最大独立集，以及相应的达成方式数即可，因此考虑状态压缩 DP。

记“已经使用了集合 S 中的点后独立集大小为 k”的方案数为 $f_{S,k}$。

考虑 DP 顺序，如果通过记录点集，再一个个加点转移，则需要记录有哪些点被加到独立集中。

所以考虑改变转移顺序，让每次被转移的点都能扩大独立集。这只需要一次性加入被转移的点及其周围的点就能做到。

然后需要安排这些点的位置：整个排列最靠前的空位必须填入被转移点，剩下位置随意填其周围没被加入过的点。

Example (「PKUWC2018」Minimax)

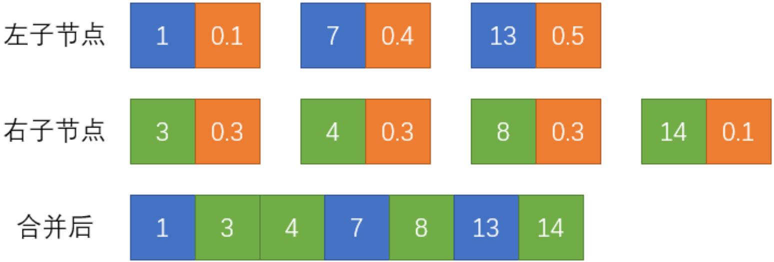
给定一棵 n 个点的有根树，定义点 x 的权值为：

- 如果 x 没有子节点，那么其权值在输入给出（保证输入的权值互不相同）。
- 如果 x 有子节点，那么其权值有 p_x 的概率是它子节点权值的最大值，有 $1-p_x$ 是子节点权值的最小值。

对于根节点权值所有可能的结果求出其概率。
数据范围 $n \leq 3 \times 10^5, w_i \leq 10^9, p_x > 0$。

先考虑简单情况，假设一个节点至多有两个子节点。

对于一个有子节点的节点，在计算出其子节点可能的权值后，其权值集合是子节点权值的并集。



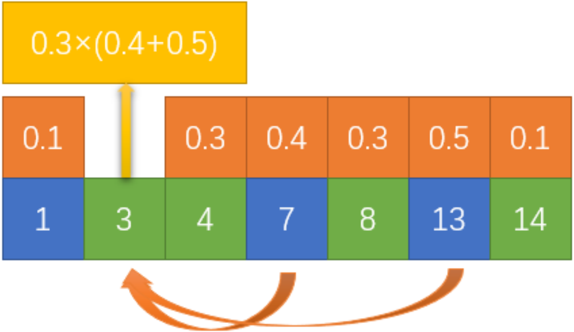
容易发现，对于有子节点的节点 x，取最小和取最大的情况类似，所以只讨论取最小值。

在枚举 x 的权值是 v 时，这个 v 一定对应了其中一个子节点 y。

这对应了 y 的权值是 v，所以只需要枚举另一个子节点 z 的权值就可以枚举出 x 的权值是 v 时所有的情况。

这里只要求 z 中选的权值是大于 v 的，最终概率为 y 选 v 的概率乘上 z 选 $> y$ 的概率。

这里只要求 z 中选的权值是大于 v 的，最终概率为 y 选 v 的概率乘上 z 选 $> y$ 的概率。



这里就得出了一个 $O(n^2)$ 的树形 DP 做法，因为考虑权值列表的两两合并，每对点只会在其最近公共祖先处产生合并。

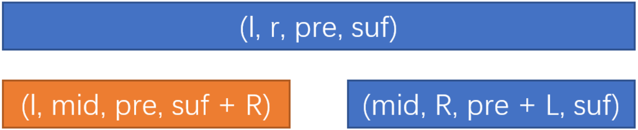
对于多叉树，可以用上述方法将子节点依次合并。

每个节点合并的结果是 p_x 倍的按最大值合并的结果，加上 $1-p_x$ 倍的按最小值合并的结果。

所以上述过程很容易地拓展到了原题上。

但是 n 很大，$O(n^2)$ 远远不够，这就要用到线段树合并。

在上述的计算中，利用到了前后缀的信息，而线段树的分治过程中正好可以维护：



通过动态开点的线段树，以权值为键以概率为值存状态，线段树节点数的性质保证了最终复杂度。

在合并的过程中，同时维护 $[0, l)$ 和 $(r, +\infty)$ 的部分和信息，即可完成转移。至此计算过程被优化到了 $O(n \log n)$ 时间复杂度。

Example (「PKUWC2018」猎人杀)

有 n 个猎人，每个猎人有 w_i 的仇恨度。每个猎人死后必须开一枪，并且被射中的人也会死。

每个猎人死后，假设活着的猎人有 $[i_1, \ldots, i_m]$，那么有 $\dfrac{w_{i_k}}{\sum_{j=1}^{m} w_{i_j}}$ 的概率向猎人 i_k 开枪。

按照仇恨度带权随机向一个猎人开枪以开始。由于开枪的连锁反应会导致最终所有猎人死亡，求 1 号猎人最后一个死的概率。

数据范围 $1 \leq \displaystyle\sum_{i=1}^{n} w_i \leq 10^5$。答案模 998244353。

棘手之处是活着的猎人列表会变，难以直接处理。
注意到题目中求 1 号猎人最后一个死的概率，可以发现：

- 如果开枪时不考虑目标是否活着，而是任意开枪，打到死人死人也会接着开枪，那么答案不会变。
- 因为这和挑到一个活人开枪的概率是一样的。
- 这个概率相同是因为最后总会有个死人打活人，而打活人只在活人列表里随机。

这使得我们可以将被开枪序列抽象为，前 k 次打编号 $2 \dots n$ 里的猎人，第 $k+1$ 次打编号为 1 的猎人。

序列抽象为，前 k 次打编号 $2\dots n$ 里的猎人，第 $k+1$ 次打编号为 1 的猎人。

但是会有一个问题，有可能这前 k 次存在猎人一次没被打到，那么 1 就不可能最后一个死。

为了处理这种情况，使用容斥原理计算（令 $W=\sum\limits_{i=1}^{n} w_i$）：

$$Ans=\sum_{S\subseteq\{2,\dots,n\}}(-1)^{|S|}\sum_{k=0}^{+\infty}\left(\frac{W-w_1-\sum_{i\in S}w_i}{W}\right)^k\frac{w_1}{W}.$$

注意到后者是一个等比数列求和，其值只与 $\sum\limits_{i\in S} w_i$ 相关，还会带上 $(-1)^{|S|}$ 的容斥系数。所以只需要求出多项式 $\prod\limits_{i=2}^{n}(1-x^{w_i})$ 即可。使用分治 FFT 即可做到 $O(W\log^2 W)$。

Definition (条件概率)

设事件 $A, B \in \mathcal{F}$ 且 $P(A) > 0$，记

$$P(B|A) = \frac{P(AB)}{P(A)}$$

为已知事件 A 发生条件下，事件 B 发生的条件概率。

条件概率将概率限制在了一个新的概率空间中。

根据条件概率可以写出条件概率的乘法公式：

Theorem (条件概率的乘法公式)

对于 $A, B \in \mathcal{F}$，若 $P(A) > 0$，那么：

$$P(AB) = P(A)P(B|A).$$

可以做以下拓展

Theorem (条件概率的一般乘法公式)

对于 $A_i \in \mathcal{F}$，若 $P(A_1A_2\dots A_n) > 0$，那么：

$$P(A_1A_2\dots A_n) = P(A_1)P(A_2|A_1)\dots P(A_n|A_1A_2\dots A_{n-1}).$$

Definition (完备事件群)

称有限多个或者无穷可列多个事件 $\{A_i\}$ 为 Ω 的完备事件群，如果：

1. 对于任意 A_i，有 $A_i \in \mathcal{F}, P(A_i) > 0$,
2. $\bigcup\limits_i A_i = \Omega$,
3. A_i 两两不交。

即 A_i 构成 Ω 的一组分割。

Theorem (全概率公式)

对于 $B \in \mathcal{F}$ 以及完备事件群 $\{A_i\}$，有：

$$P(B) = \sum_i P(A_i)P(B|A_i).$$

全概率公式.

$$B = B\Omega = B\left(\bigcup_i A_i\right) = \bigcup_i BA_i,$$

$$P(B) = P\left(\bigcup_i BA_i\right) = \sum_i P(BA_i) = \sum_i P(A_i)P(B|A_i).$$

□

$$P(B)=\sum_i P(A_i)P(B|A_i).$$

全概率公式意思就是，在计算一个事件的概率时，可以使用一个概率空间的划分来枚举该事件的所有前提。

Theorem (贝叶斯公式 (Bayes' theorem))

对于 $B \in \mathcal{F}$ 以及完备事件群 $\{A_i\}$，有：

$$P(A_i|B)=\frac{P(A_iB)}{P(B)}=\frac{P(A_i)P(B|A_i)}{P(B)}=\frac{P(A_i)P(B|A_i)}{\sum_k P(A_k)P(B|A_k)}$$

Example (三门问题)

有三扇门，其中一扇背后有奖，剩下两扇没奖，你需要挑出那一扇有奖的门。
当你选定了一扇门后，主持人会打开另一扇没有奖的门。在这之后你有一次改变选择的机会，请问改变选择会增加最后获奖的概率吗？

Example (三门问题)

有三扇门，其中一扇背后有奖，剩下两扇没奖，你需要挑出那一扇有奖的门。
当你选定了一扇门后，主持人会打开另一扇没有奖的门。在这之后你有一次改变选择的机会，请问改变选择会增加最后获奖的概率吗？

假设一开始选的门为 A，被开的门为 B，剩下门 C。

设 Y 为选出 B 门没奖的事件，Z 为 C 门有奖的概率。因为概率空间限制于选了 A 门之后，则由贝叶斯公式：

$$P(Z|Y) = \frac{P(Z)P(Y|Z)}{P(Y)} = \frac{\frac{1}{3}\cdot 1}{\frac{1}{2}} = \frac{2}{3}$$

则改变选择真的会增加获奖的概率。

Example (N 门问题)

Alice 来参加节目，主持人给出 n 扇门，其中一扇门有奖。

当 Alice 选择一扇门但不打开，主持人会在所有门除了 Alice 目前选择的以及有奖的门中随机开一扇门打开。然后 Alice 会在所有没开的门中再次选择一扇她认为获奖概率最高的（如果有多个就等概率随机选择一扇），然后主持人重复操作……

这样两人不断选择知道剩下两个门，Alice 会选择一扇概率更高的门作为最终选择。

现在 Bob 来当主持人暗箱操作，他会选择性地开门使得 Alice 最终中奖概率最小。而 Alice 不知道主持人在针对她，认为主持人等概率随机开门。

问 Bob 在使用最佳策略时，Alice 获奖的概率。

数据范围：$n \le 10^{18}$。

先考虑理解题意，设计一个暴力枚举过程：Alice 内心有个对于每个门的概率表。

每次 Alice 选择概率最大的那个（若有多个可以枚举），然后 Bob 枚举选择哪个没奖的展示。

在 Bob 展示后因为 Alice 认为 Bob 是随机选的，所以 Alice 会使用条件概率来计算各门新的概率。

这样的过程是可以递归的，也就是可以通过 dfs 知道对于 Bob 来说什么样是最优的决策，以计算出 Alice 获奖的概率。

对于一轮中，设 X_i 为第 i 个门有奖，并假设 1 是 Alice 这轮开的门。

设 Z 为 Bob 在目前能选的门等概率随机选出了一个没有奖的。假设现在有 m 个门没开（在 Bob 这轮开之前）。

之前发生的所有事件将概率限定在了概率空间 $(\Omega, \mathcal{F}, P_Y)$ 中，根据贝叶斯公式：

$$P_Y(X_i|Z) = \frac{P_Y(X_i)P_Y(Z|X_i)}{\sum_k P_Y(X_k)P_Y(Z|X_k)}$$

而 $P_Y(Z|X_i) = \dfrac{1}{m-2+[i=1]}$，于是在 Bob 开门后 Alice 内心的概率表可以通过之前的概率表得到。

但是搜索太慢了，对于 $n \le 10^{18}$ 根本没办法跑，尝试优化。

注意到 Alice 概率表的变化实际上是选择一个概率最大的门，将其概率变小（乘上一个系数）。由归纳知，其实它是最小的。

如果剩下两个门时，正确的门正好是概率最小的，也就是控制正确的门被 Alice 选的时间，就可以让 Alice 失败。

如果 n 足够大，因为概率表元素相对顺序的改变量是两倍于操作的，那么 Bob 可以操控正确门相对概率更大或更小门的数量，根据奇偶性等给 Alice 最终安排到错误的门。（因为 Bob 可以选门删掉，这使得有简单的策略实现这个）

经过打表试验，当 $n > 10$，Alice 就完全赢不到了。

（据考场上的选手说，打表找规律，然后暴力跑着跑着发现 $n > 10$ 答案就变成 0 了）

随机变量与数学期望

Definition (示性函数)

设 $(\Omega, \mathcal{F}, P)$ 为概率空间，对于事件 $A \in \mathcal{F}$，其示性函数为：

$$I_A(\omega) = \begin{cases} 1, & \omega \in A, \\ 0, & \omega \notin A. \end{cases}$$

Definition (随机变量)

称样本空间 Ω 上的实值函数 $X : \Omega \to \mathbb{R}$ 为 $(\Omega, \mathcal{F})$ 上的随机变量。

Definition

对随机变量定义运算

$$X \le x := \{\omega \in \Omega | X(\omega) <= x\}$$

由事件体 $\mathcal{F}$ 对至多可数无穷多次交并补运算是封闭的，所以可以定义：

$$\begin{aligned} X < x :=& \bigcup_{n=1}^{+\infty} \left(X \le x - \frac{1}{n} \right), \\ X = x :=& (X \le x) - (X < x), \\ X \ge x :=& \Omega - (X < x). \end{aligned}$$

一般的，对于 $S \subseteq \mathbb{R}$, 有

$$X \in S := \{\omega \in \Omega | X(\omega) \in S\}$$

离散型随机变量和连续型随机变量

Definition (离散型随机变量、分布与分布列)

若一个随机变量可以去至多可数无穷多个值，则称之为离散型随机变量（discrete random variable）。
设 x_i 是离散型随机变量所有可能的取值，令 $p_i = P(X = x_i)$，则称 $\{p_i\}$ 为 X 的分布，称矩阵

$$\begin{bmatrix} x_1 & x_2 & \dots & x_n & (\dots) \\ p_1 & p_2 & \dots & p_n & (\dots) \end{bmatrix}$$

为 X 的分布列。

Definition (连续型随机变量与概率密度函数)

对于随机变量 X，若存在可积函数 $f(x)$ 使得

$$P(X \le x) = \int_{-\infty}^{x} f(x)\mathrm{d}x,$$

则称 X 为连续型随机变量（continuous random variable），称这个 $f(x)$ 为 X 的概率密度函数（probability density function）。

概率密度函数对初学者可能有点反直觉，其意思是 $\mathrm{d}x$ 区间内有 $f(x)\mathrm{d}x$ 的概率。

除了这两种随机变量，还有混合型的，这里不再介绍。

分布函数

Definition (分布函数)

设 X 为概率空间 $(\Omega, \mathcal{F}, P)$ 上定义的随机变量，称

$$F_X(x) := P(X \leq x)$$

为 X 的分布函数（distribution function）。

随机变量与分布函数用实值函数刻画了随机现象，

Definition (数学期望)

设随机变量 X 的分布函数为 $F_X(x)$，定义其数学期望（expectation）为（若积分绝对可积）：

$$E[X] = \int_{-\infty}^{+\infty} x \, \mathrm{d}F_X(x).$$

对于离散型随机变量，该积分为：

$$E[X] = \sum_i x_i p_i.$$

对于连续性随机变量，该积分为：

$$E[x] = \int_{-\infty}^{+\infty} x \, f_X(x) \mathrm{d}x.$$

期望在 OI 的概统题中极其常见，大量的题是让你求期望的。

Property (期望的性质)

1. 对于常函数 C，$E[C] = C$.
2. 线性性：$E[\lambda X + \mu Y] = \lambda E[X] + \mu E[Y]$.
3. 给定两两独立的 $X_1, \dots, X_n$，有 $E[X_1 X_2 \dots X_n] = \prod_{i=1}^{n} E[X_i]$.
4. $\forall x \in \mathbb{R}, E[(X - EX)^2] \leq E[(X - x)^2]$.

线性性尤其重要，经常被拿来解题。

Example

在游戏 Noita 中有一个物品叫闪耀魔球，你踢一脚有以下情况可能发生：

1. $1/26$ 概率，你不会获得金钱。
2. $8/26$ 概率，你会获得 10 金钱。
3. $15/26$ 概率，你会获得 20 金钱。
4. $1/26$ 概率，你会获得 100 金钱。
5. $1/26$ 概率，你会获得 200 金钱。

问踢一脚期望获得多少金钱？

构造离散型随机变量 $X(\omega)$ 使其值为事件对应金钱数，则期望为：

$$E[X] = \frac{1}{26} \cdot 0 + \frac{8}{26} \cdot 10 + \frac{15}{26} \cdot 20 + \frac{1}{26} \cdot 100 + \frac{1}{26} \cdot 200 = \frac{340}{13} \approx 26.154.$$

Example

一个 k 面骰子，不断骰直到骰到 1，骰的次数的期望是 k。

Example

一个 k 面骰子，不断骰直到骰到 1，骰的次数的期望是 k。

考虑事件 A_i 表示第 i 次才骰到 1，其概率为 $\left(\frac{k-1}{k}\right)^{i-1}\frac{1}{k}$。

构造离散型随机变量使得 $X(A_i)=i$。那么期望次数是：

$$E[X]=\sum_{i=1}^{+\infty} i\left(\frac{k-1}{k}\right)^{i-1}\frac{1}{k}=\frac{1}{k}\left(\sum_{i=1}^{+\infty} ix^{i-1}\right)\quad,x\to\frac{k-1}{k}.$$

其中 $\sum_{i=1}^{+\infty} ix^{i-1}=\frac{\mathrm{d}}{\mathrm{d}x}\frac{1}{1-x}=\frac{1}{(1-x)^2}$, 则期望为 $\frac{\frac{1}{k}}{\frac{1}{k^2}}=k.$

Example (NOIP 2018 提高组初赛第 7 题)

在一条长度为 1 的线段上随机取两个点，则以整两个点为端点的线段期望长度是多少？

Example (NOIP 2018 提高组初赛第 7 题)

在一条长度为 1 的线段上随机取两个点，则以整两个点为端点的线段期望长度是多少？

首先考虑事件取到长度为 l 的线段，设其权重为 $1-l$。
则通过带权平均算出每个事件发生的概率密度：
$\int_0^1 (1-l)\mathrm{d}l = \frac{1}{2}$，则有概率密度函数 $f(l)=2(1-l)$.
直接通过数学期望定义的积分：

$$\int_0^1 2l(1-l)\mathrm{d}l = \left(l^2 - \frac{2}{3}l^3\right)\bigg|_{l=0}^{l=1} = \frac{1}{3}.$$

Example (绿豆蛙的归宿)

给定一张有向无环图，给定唯一起点与唯一终点，保证起点可以到达所有点，所有点可以到达终点。
每条边有个长度，你需要从起点出发走到终点。
在离开每个点时，假设这个点有 k 条出边，则会在这些出边里等概率选取一条边走（即概率为 $\frac{1}{k}$）。求从起点走到终点的路径总长度数学期望值。
要求线性时间复杂度。

为每条边 e 设一个随机变量 $X_e(\omega)$，当这条边被经过则 $X_e(\omega) = 1$ 没有则 $X_e(\omega) = 0$。

此时由期望线性性，可以将所有这些随机变量组合起来，即计算 $E\left[\sum_e X_e\right]$ 得到题目所有的路径总长度。

此时问题被转化成计算每条边被经过的概率 $P(A_e)$，$P(A_e)$ 可以由点被经过概率 $P(B_v)$ 计算，即 $A_e = \dfrac{P(B_v)}{d^+(v)}$。

此时问题被转化成计算每条边被经过的概率 $P(A_e)$，$P(A_e)$ 可以由点被经过概率 $P(B_v)$ 计算，即 $A_e = \dfrac{P(B_v)}{d^+(v)}$。

由有向无环图性质，对于每个点，其所有前趋边被经过的事件以及“其任何前趋都没有被经过”的事件构成了完备事件群。

根据全概率公式，记前趋边集合为 F，有

$$
\begin{aligned}
P(B_v) &= \sum_{e\in F} P(B_v|A_e)P(A_e) + P(B_v|\Omega - \cup_{e\in F}A_e)P(\Omega - \cup_{e\in F}A_e) \\
&= \sum_{e\in F} P(A_e).
\end{aligned}
$$

按照拓扑序递推即可。

Example (红包发红包)

给一个实数 w_0，每轮等概率随机一个 $[0, w_{i-1}]$ 间的实数 x_i，令 $w_i = w_{i-1} - x_i$。问 x_k 的期望大小。

Example (红包发红包)

给一个实数 w_0，每轮等概率随机一个 $[0, w_{i-1}]$ 间的实数 x_i，令 $w_i = w_{i-1} - x_i$。问 x_k 的期望大小。

初始的数对答案是线性影响的，也就是每次随机对结果的期望都是独立贡献的，也就是做乘积。

所以答案就是 $\frac{w_0}{2^k}$。

Example (WJMZBMR 打 osu! / Easy)

给一个由 $o, x, ?$ 三种字符组成的字符串。等概率随机往 ? 里填 o 或 x。
一个字符串的得分是所有极大连续 o 的长度平方和，例如
$ooxxxxooooxxx$ 的得分是 $2^2 + 4^2 = 20$。
求等概率随机填入的情况下，得分的期望值。
数据范围：字符串长度 $\leq 3 \times 10^5$。

如果直接 DP，则需要记录当前连续 o 长度，显然是不可接受的。
考虑 x^2 如何递推地计算：$(x+1)^2 = x^2 + 2x + 1$。
也就是说，只要在 DP 时，对前 i 个字符，存下：

1. 后缀为 o 的概率，
2. 后缀最长连续 o 长度和期望，
3. 后缀最长连续 o 长度平方和期望。

即可递推地转移出 x^2 的期望。

实际上根据其组合意义，也可以写出另一种 DP：

极长连续段的平方和，实际上就是在连续段中任意选择两个元素的方案数。

这样就可以通过 $f_{i,0/1,0/1}$ 这样的状态直接计算出了。

Example (换教室)

给定一张无向带权连通图表示教室，你需要按顺序上 n 节课。每节课在 a_i 教室上，如果提出申请换教室，则有 k_i 的概率可以通过，如果通过了你可以在 b_i 教室上。
各节课之间你会走最短路，花费是相邻课教室间路径长度之和。即从第一节课的教室走到第二节的，再走到第三节的……
你可以提出至多 m 个申请。最小化你走的路程期望。
数据范围：$n, m \leq 2000$，图点数 ≤ 300。

预处理最短路后，只需要 $O(nm)$ 的 DP 设计即可。

很容易设计这样的 DP：对于前 i 节课，记录提出了多少申请，以及最后一个教室是否换过的情况下，路径长度期望值最小是多少。

在转移的时候，根据相邻两次是否换教室，可以根据换教室是否成功写出距离的期望公式。

由于期望的线性性，这样累加地转移得到的结果是正确的。

Example (游走)

给定一张无向连通图，有 n 个点 m 条边。
你初始在 1 号点，每一步会等概率随机一条相邻的边并走到另一端点，直到走到 n 号点。
现在你需要给边编上 1 到 m 的权值，最大化这个过程中你经过边的期望权值和。
数据范围：$n \leq 500$。

很明显，这是需要你求出每条边被经过的期望次数 E_e。先设出每个点被经过的期望次数 E_v，可以列出以下方程（设 W 是 v 的相邻点）：

$$
\begin{aligned}
E_v =& [v=1] + \sum_{w \in W} [w \neq n] \frac{E_w}{d(w)} \\
E_e =& [u \neq n] \frac{E_u}{d(u)} + [v \neq n] \frac{E_v}{d(v)} \quad , e = (u, v)
\end{aligned}
$$

现在的结构已经无法像之前那样使用递推求出答案。

注意到 $n \le 500$，这说明答案可以直接用高斯消元等解方程手段求出，然后按照 E_e 排序给边赋权。

为什么可以把期望直接列到方程里？

如果你设出第 k 步在每个点的概率来算期望的话，可以发现期望值间的关系恰好满足这个方程。

而且如果图连通，这个方程是有唯一解的，这说呢这个方程的解恰好对应正确的期望值。

Example (树上随机游走)

给定一棵无根树，对于每个点求，从这个点出发，每次等概率选择所在点周围一条边走到另一端点，期望走多少步能走到叶子上。
要求线性时间复杂度（不包含求逆元）。

Example (树上随机游走)

给定一棵无根树，对于每个点求，从这个点出发，每次等概率选择所在点周围一条边走到另一端点，期望走多少步能走到叶子上。
要求线性时间复杂度（不包含求逆元）。

首先得到“从这个点出发期望走多少步走到叶子上”的方程（对于叶子有 $E_v = 0$，设 W 为 v 相邻点的集合）：

$$E_v = \sum_{w \in W} \frac{1}{d(w)} E_w + 1.$$

在线性时间复杂度情况下难以使用高斯消元，但是变量之间的依赖使得递推也无法使用。

为了带来点求解的顺序，为这棵树指定一个根。

$$E_v = \frac{1}{d(v)} \sum_{w \in W} E_w + 1.$$

尝试来模拟这个消元过程。注意一些特殊的节点：那些子节点只有叶子的节点。因为叶子的期望为 0，也就是这个点的期望值只和其父亲的期望相关，可以表示为 $E_v = a_v E_f + b_v$。

设叶子点集合为 V_0，这些特殊点集合为 V_1，则可以继续寻找子节点只有 $V_0 \cup V_1$ 的点作为 V_2。

对于 V_2 中的点，将其在 V_1 中的子节点代入自己的方程（即 $E_v = cE_v + aE_f + b$），这时候可以移项把等式右边的 c 消掉，发现该点也变成了 $E_v = a_v E_f + b_v$ 的形式。

也就是说可以通过把子树全部变成 $E_v = a_v E_f + b_v$ 的形式，将自己也变成这个形式。使用 dfs 完成。

注意到根节点是没有父亲的，也就是计算到根节点的时候，根节点的期望步数已经确定了。

这个时候就可以从根到叶子逐步代入计算出每个点的期望了。

Example (分手是祝愿)

有 n 个灯和 n 个开关，给出每个灯是否亮着。
每次按第 i 个开关的时候，所有 i 的约数的灯状态会被反转。
为了使所有灯灭掉，执行若干操作：

- 每一次操作，都等概率随机选择一个开关来操作。
- 直到存在一种方案使得可以操作小于等于 k 个开关使所有灯灭掉，
- 此时直接操作 k 次完成目标。

问期望多少次操作可以灭掉所有灯。
数据范围：$k \le n \le 10^5$。

首先注意到操作顺序是可交换的。同时对于一组灯的状态，最快的关灯方式对应唯一集合。

并且注意到最快操作次数是有上限的，所以现在的问题就变成了一个链上的随机游走问题。

对于 a 次操作的状态，有 $\frac{a}{n}$ 的概率向 $a-1$ 走，$\frac{n-a}{n}$ 的概率向 $a+1$ 走。

有边界条件 $E(X_k)=k$。

直接按类似上一题的做法计算，或者直接按照操作数从小到大代入法解方程也可以。

Example (大鱼洽水)

给定一棵边仙人掌（每条边至多在一个简单环上），从一个点出发，每次等概率选择所在点周围一条边走到另一端点。
对每个点求，如果这个点作为起点，期望走多少步能走到指定点集的任意一点上？要求线性时间复杂度（不包含求逆元）。

考虑和树上随机游走类似的做法，现在需要额外解决环上随机游走的问题。

环上变成了 $f_v = a_v f_{L_v} + b_v f_{R_v} + c_v$。

如果指定了环的根 w，可以将 w 看作类似于“环的父节点”的角色（在“点双连通分量”中也是如此）。

而这个 w 一般下面挂着多个环，比较复杂。所以在环的计算中尝试规避。将其值设为未知数 x。

将 x 代入 w 下面挂着的环，环中元素失去了与环根的依赖，成了条链。使用树上的做法能够轻松将环两端点的方程表示成仅与 x 相关的式子。

将所有这样两端点的式子代入 x 的方程，即可将 x 化为与父节点或父环相关的形式。递推到根后将值逐个代入即可计算出答案。