

概率和期望题选讲

上海交通大学 方泓杰

概念与性质

$$E = \sum_i p_i w_i, \left(\sum_i p_i = 1 \right)$$

p_i 为事件 i 发生概率， w_i 为事件 i 发生收益， E 为收益期望。

期望线性性： $E(ax + by + cz) = aE(x) + bE(y) + cE(z)$ (a, b, c 为常数, x, y, z 为变量)

期望的积不等价于积的期望！！

即： $E(xy)$ 和 $E(x)E(y)$ 无直接关系！！

常用套路

一般：动态规划，写出转移方程。

如果转移方程成环，考虑高斯消元或分离系数。

需要有一定的公式推导能力。

有时候，求期望可能直接可以求出每种情况的方案数，然后用期望定义得到答案。

收集邮票 Minus

有 n 种不同的邮票，皮皮想收集所有种类的邮票。

每次只能买一张，并且买到的邮票是 n 种的哪一种是等概率的，概率均为 $\frac{1}{n}$ 。

皮皮购买每张邮票要支付1元钱。

现在皮皮手中没有邮票，皮皮想知道自己得到所有种类的邮票需要花费的钱数目的期望。

$$1 \leq n \leq 10^4$$

收集邮票 Minus

令 $g(i)$ 表示现在有 i 张，要买到 n 张的期望次数

那么，又有：

$$g(i) = 1 + g(i) \times \frac{i}{n} + g(i+1) \times \frac{n-i}{n}$$

化简有

$$g(i) = g(i+1) + \frac{n}{n-i}$$

递推即可，时间复杂度 $O(n)$ 。

Kids and Prizes

有 n 个奖品， m 个人排队选礼物。

对于每个人，他打开的盒子可能有礼物，也可能已经被之前的人取走了。

如果有礼物，取走礼物然后把盒子放回原处。

求最后所有人期望取走多少个礼物。

$$1 \leq n, m \leq 10^5$$

SGU 495 Kids and Prizes

设 $f(i)$ 表示前 i 个人取走礼物个数的期望，那么

$$f(1) = 1, f(i) = f(i-1) + \frac{n - f(i-1)}{n} \times 1 + \frac{f(i-1)}{n} \times 0 = f(i-1) + \frac{n - f(i-1)}{n}$$

时间复杂度 $O(n)$ 。

Help Me Escape

有 n 个妖怪，每个妖怪有一个战斗力 c_j 。小A有一个初始战斗力 f_0 。

每天，小A随机选择一个妖怪决斗。设小A在第 i 天的战斗力为 f_i 。

设当前为第 i 天，如果打赢，即 $f_i > c_j$ ，就可以逃出去，需要花 t_j 天。

如果打不赢，小A会让自己变强， $f_{i+1} = f_i + c_j$ 。

求小A逃出来需要的期望天数。

$$1 \leq n \leq 100, 0 \leq f \leq 10000, 0 \leq c_i \leq 10000$$

ZOJ 3640 Help Me Escape

用 $f(i)$ 表示战斗力为 i 时出去需要的期望天数。

那么：如果 $i > c_j$ ，那么 $f(i) = f(i) + \frac{1}{n}(t_j)$;

否则（ $i \leq c_j$ ），那么 $f(i) = f(i) + \frac{1}{n}(f(i + c_j) + 1)$ 。

可以记忆化搜索。

时间复杂度

Easy

某游戏有 n 次点击，给出一个长度为 n 的字符串表示点击序列。

点击成功了就是“o”，点击失败了就是“x”，暂时未知是否点击成功为“?”。

分数是按comb计算的，长度为 a 的comb就有 a^2 分，comb就是极大的连续“o”。

当标记为“?”的位置点击成功的概率为50%时，求得分期望。

$$1 \leq n \leq 3 \times 10^5$$

BZOJ 3450 Easy

分开考虑每个位置的得分期望，加在一起就是总得分期望！

设某位置之前有 q 个连续的“o”，那这个位置也是“o”的贡献 $(q + 1)^2 - q^2 = 2q + 1$ 。

设 i 位置之前的连续“o”的个数期望为 e_i ，那么这个位置期望贡献 $(2e_i + 1)p$ 。

（其中 p 为 i 这个位置取“o”的概率）

所以我们只需求出 e_i 即可。

BZOJ 3450 Easy

分开考虑每个位置的得分期望，加在一起就是总得分期望！

设某位置之前有 q 个连续的“o”，那这个位置也是“o”的贡献 $(q+1)^2 - q^2 = 2q + 1$ 。

设 i 位置之前的连续“o”的个数期望为 e_i ，那么这个位置期望贡献 $(2e_i + 1)p$ 。

（其中 p 为 i 这个位置取“o”的概率）

所以我们只需要求出 e_i 即可。

显然在已经确定“o”的位置： $e_i = e_{i-1} + 1$ ；在已经确定“x”的位置： $e_i = 0$ 。

现在只剩下“?”的位置的 e_i 还未确定。

在“?”的位置： $e_i = 0.5 \times 0 + 0.5(e_{i-1} + 1)$

时间复杂度 $O(n)$ 。

OSU!

某游戏有 n 次点击。

点击成功了就是“o”，点击失败了就是“x”。

分数是按comb计算的，长度为 a 的comb就有 a^3 分，comb就是极大的连续“o”。

给出每个位置上点击成功（“o”）的期望 p_i ，求期望得分。

$$1 \leq n \leq 3 \times 10^5, 0 \leq p_i \leq 1$$

BZOJ 4318 OSU!

类似的，先设 i 位置前连续“o”的个数为 q ，来得出某个位置的贡献：

$$(q+1)^3 - q^3 = 3q^2 + 3q + 1$$

那么，根据期望的线性性：

$$E = \sum_{i=1}^n E(i), \quad E(i) = (1 - p_i) \times 0 + p_i \times E(3q^2 + 3q + 1) = 3E_i(q^2) + 3E_i(q) + 1$$

下面考虑求 $E_i(q)$ 和 $E_i(q^2)$ 。 $E_i(q)$ 求法和前一题类似。下面即求 $E_i(q^2)$ 。

$$E_i(q^2) = (1 - p_i) \times 0 + p_i \times (E_{i-1}(q^2) + 2E_{i-1}(q) + 1) = p_i(E_{i-1}(q^2) + 2E_{i-1}(q) + 1)$$

那么即可求得答案。时间复杂度 $O(n)$ 。

Collecting Bugs

一个软件有 s 个子系统，会产生 n 种bug。

某人每天发现一个bug，这个bug属于某个子系统，属于某种bug。

每天发现的bug，出现在每个子系统概率均为 $\frac{1}{s}$ ，属于每种bug的概率均为 $\frac{1}{n}$ 。

求找到所有的 n 种bug，且每个子系统都找到bug所要天数的期望。

$$1 \leq n, s \leq 2000$$

POJ 2096 Collecting Bugs

$f(i, j)$ 表示已经找到了 i 种bug，存在于 j 个子系统中，期望还需要多少天。

那么 $f(n, s) = 0$ ， $f(0, 0)$ 即为答案。

转移方式：

1. 发现了新bug种类，存在于新子系统中，概率 $p_1 = \frac{n-i}{n} \times \frac{s-j}{s}$ ，还需要 $f(i+1, j+1)$ 步；
2. 发现了新bug种类，存在于旧子系统中，概率 $p_2 = \frac{n-i}{n} \times \frac{j}{s}$ ，还需要 $f(i+1, j)$ 步；
3. 发现了旧bug种类，存在于新子系统中，概率 $p_3 = \frac{i}{n} \times \frac{s-j}{s}$ ，还需要 $f(i, j+1)$ 步；
4. 发现了旧bug种类，存在于旧子系统中，概率 $p_4 = \frac{i}{n} \times \frac{j}{s}$ ，还需要 $f(i, j)$ 步。

那么，整理一下：

POJ 2096 Collecting Bugs

那么，根据期望的公式，

$$f(i, j) = p_1 f(i + 1, j + 1) + p_2 f(i + 1, j) + p_3 f(i, j + 1) + p_4 f(i, j) + 1$$

从而：

$$f(i, j) = \frac{p_1 f(i + 1, j + 1) + p_2 f(i + 1, j) + p_3 f(i, j + 1) + 1}{1 - p_4}$$

即得转移方程，复杂度 $O(ns)$ 。

Noip2016 换教室

给出一张无向图 $G = (V, E)$ 作为学校地图。

小A选了 n 门课，这 n 门课依次上课。

现在他可以更换 m 门课，每门课原来在点 c_i 上课，更换后在 d_i 上课，更换成功概率为 p_i 。

从一个教室到另一个教室的体力消耗为两个教室之间的距离。求最小期望体力消耗。

$$1 \leq n, m \leq 2000, 1 \leq V \leq 300, 1 \leq E \leq 90000$$

Noip2016 换教室

首先可以Floyd预处理出点对之间的最短路。

然后考虑dp，令 $f_{i,j}$ 表示前 i 个课程，选择了 j 个要换的，期望的体力消耗最小值。

如何转移？

Noip2016 换教室

首先可以Floyd预处理出点对之间的最短路。

然后考虑dp，令 $f_{i,j}$ 表示前 i 个课程，选择了 j 个要换的，期望的体力消耗最小值。

如何转移？

这样没办法表示上一次是否选择了要换。

所以我们需要更改状态表示：

$f(i, j, 0/1)$ 表示前 i 个课程，选了 j 个要换的，其中最后一个是否换，期望的体力消耗最小值

Noip2016 换教室

那么转移:

$$\begin{aligned} f_{i-1,j,0} &\rightarrow f_{i,j,0}, f_{i-1,j,1} \rightarrow f_{i,j,0} \\ f_{i-1,j-1,0} &\rightarrow f_{i,j,1}, f_{i-1,j-1,1} \rightarrow f_{i,j,1} \end{aligned}$$

考虑每个是否要换即可。

时间复杂度 $O(nm + V^3)$ 。

摘苹果

在花园中有 n 棵苹果树以及 m 条双向道路，苹果树编号依次为1到 n 。

每条道路的两端连接着两棵不同的苹果树，假设第 i 棵苹果树连接着 d_i 条道路。

小Q将会按照以下方式去采摘苹果：

1. 小Q随机移动到一棵苹果树下，移动到第 i 棵苹果树下的概率为 $\frac{d_i}{2m}$ ，但并不在此采摘。
2. 等概率随机选择一条与当前苹果树相连的一条道路，移动到另一棵苹果树下。
3. 设当前位于第 i 棵苹果树下，则他会采摘 a_i 个苹果，多次经过同一棵苹果树下会重复采摘。
4. 重复第2步和第3步 k 次。请写一个程序帮助计算小Q期望摘到多少苹果。

对 $(10^9 + 7)$ 取模。

$$1 \leq n, k \leq 10^5, 1 \leq m \leq 2 \times 10^5$$

BZOJ 5091 摘苹果

一开始停留在某个点的概率为 $\frac{d_i}{2m}$ ，在经过一次操作后停留在某个点的概率为：

$$p_i = \sum_{(j \rightarrow i)} \frac{d_j}{2m} \times \frac{1}{d_j} = \frac{d_i}{2m}$$

不发生变化！说明和次数无关。

利用期望的线性性可求得期望：

$$E = \sum_{i=1}^n E(i) = \sum_{i=1}^n k a_i p_i = \frac{k}{2m} \sum_{i=1}^n a_i d_i$$

总复杂度 $O(n)$ 。

Bad Luck Island

在一个岛上，有 $(r + s + p)$ 个人，其中有 r 个人有石头， s 个人有剪刀， p 个人有布。

遵循石头剪刀布的原则，输的人就狗带了。每两个人遇到概率的相等。

求每个人存活的概率。

显然有同种东西的人存活概率相等，你只需要输出有石头/剪刀/布的人的存活概率即可。

$$1 \leq r, s, p \leq 100$$

CodeForces 540D Bad Luck Island

$f(i, j, k)$ 表示岛上还有*i*个石头，*j*个剪刀，*k*个布的概率。

那么初始 $f(r, s, p) = 1$ ，转移：

$$f(i-1, j, k) = f(i-1, j, k) + f(i, j, k) \times \frac{ik}{ij+ik+jk}$$

$$f(i, j-1, k) = f(i, j-1, k) + f(i, j, k) \times \frac{ij}{ij+ik+jk}$$

$$f(i, j, k-1) = f(i, j, k-1) + f(i, j, k) \times \frac{jk}{ij+ik+jk}$$

答案怎么算？

CodeForces 540D Bad Luck Island

$f(i, j, k)$ 表示岛上还有*i*个石头，*j*个剪刀，*k*个布的概率。

那么初始 $f(r, s, p) = 1$ ，转移：

$$f(i - 1, j, k) = f(i - 1, j, k) + f(i, j, k) \times \frac{ik}{ij+ik+jk}$$

$$f(i, j - 1, k) = f(i, j - 1, k) + f(i, j, k) \times \frac{ij}{ij+ik+jk}$$

$$f(i, j, k - 1) = f(i, j, k - 1) + f(i, j, k) \times \frac{jk}{ij+ik+jk}$$

答案怎么算？

如果最后只有石头/剪刀、剪刀/布、布/石头，那么胜负显然，直接统计即可。

时间复杂度 $O(rsp)$ 。

Kleofáš and the n-thlon

有 m 个人参加 n 项全能比赛。每场比赛每个人有一个比赛得分，在 $[1, m]$ 之间。

对于同一场比赛，不存在两个人得分相同。

总得分为每场比赛得分相加。

现在有一个人，他只记得他自己每场比赛的得分 c_i 。

求这个人最后排名的期望值。

$$1 \leq n \leq 100, 1 \leq m \leq 1000$$

CodeForces 601C Kleofáš and the n-thlon

$f(i, j)$ 表示前 i 项比赛，得分为 j 的期望人数。那么，转移：

$$f(i, j) = \frac{1}{m-1} \sum_{k=j-m}^{j-1} f(i-1, k) \quad (k \neq j - c_i)$$

转移是一个前缀和的形式，可以 $O(1)$ 转移。

答案显然为 $1 + \sum_{t=1}^{C-1} f(n, t)$ （其中， $C = \sum_{i=1}^n c_i$ ）

那么dp的复杂度就是状态的复杂度 $O(n^2m)$ 。

Bag of mice

袋子里初始有 w 只白鼠和 b 只黑鼠，小A和小B轮流从袋子里抓老鼠。

谁先抓到白鼠谁就赢。

小A每次抓一只老鼠（不放回）；

小B每次抓一只老鼠（不放回），但是抓完后会有另外一只老鼠溜出来。

每次抓出来/溜出来的老鼠随机。

小A先抓，求小A胜利的概率。

$$0 \leq w, b \leq 1000$$

CodeForces 148D Bag of mice

$f(i, j)$ 表示现在有 i 只白鼠， j 只黑鼠，小A赢的概率。

分类：

1. 直接选到白鼠： $\frac{i}{i+j}$;

2. 小A选到黑鼠，那么小B必须选黑鼠。

(1) 溜出来白鼠： $\frac{j}{i+j} \times \frac{j-1}{i+j-1} \times \frac{i}{i+j-2} \times f(i-1, j-2)$;

(2) 溜出来黑鼠： $\frac{j}{i+j} \times \frac{j-1}{i+j-1} \times \frac{j-2}{i+j-2} \times f(i, j-3)$ 。

从而
$$f(i, j) = \frac{i}{i+j} + \frac{j}{i+j} \times \frac{j-1}{i+j-1} \times \frac{i}{i+j-2} \times f(i-1, j-2) + \frac{j}{i+j} \times \frac{j-1}{i+j-1} \times \frac{j-2}{i+j-2} \times f(i, j-3)$$

复杂度 $O(wb)$ 。可以用记忆化搜索实现。

亚瑟王

有一套 n 张带技能的卡牌。第 i 张卡牌的技能发动概率为 p_i ，如果成功发动，则会对敌方造成 d_i 点伤害。

共有 r 轮。每轮将从第一张卡牌开始，按照顺序依次考虑每张卡牌，没有特殊情况的话考虑完所有卡牌即一轮结束。在一轮中，对于依次考虑的每张卡牌：

如果这张卡牌在这一局游戏中已经发动过技能，跳过之。

否则（这张卡牌在这局游戏中没有发动过技能），设这张卡牌为第 i 张，将以 p_i 概率发动技能。

如果技能发动，则对敌方造成 d_i 点伤害，并直接结束这一轮。

求出这一套卡牌在一局游戏中能造成的伤害的期望值。多组数据。

$$1 \leq T \leq 444, 1 \leq n \leq 220, 0 \leq r \leq 132, 0 < p_i < 1, 0 \leq d_i \leq 1000$$

BZOJ 4008 [HNOI2015] 亚瑟王

由于每轮可以直接在发动技能后结束，所以我们不能以轮次为状态考虑问题。

由于一张牌只能发动一次技能，所以我们以**机会**为单位考虑问题。

也就是：给一张卡牌多少次机会来发动技能。

设 $f_{i,j}$ 为给第 i 张牌分配 j 次机会的概率。

首先，如果第 i 张牌有 j 次机会，那么第 $i-1$ 张牌至少有 j 次机会。那么考虑转移：

1. 第 $i-1$ 张牌用掉了一次机会： $f(i-1, j+1) \times (1 - (1 - p_{i-1})^{j+1})$

注：后面的式子可以从等比数列求和推得（枚举哪一次用掉了机会）

2. 第 $i-1$ 张牌没有用掉机会： $f(i-1, j) \times (1 - p_{i-1})^j$

总方程： $f(i, j) = f(i-1, j+1) \times (1 - (1 - p_{i-1})^{j+1}) + f(i-1, j) \times (1 - p_{i-1})^j$

BZOJ 4008 [HNOI2015] 亚瑟王

那么答案就很好算了：

$$ans = \sum_{i=1}^n \sum_{j=1}^r f(i, j) \times (1 - (1 - p_i)^j) \times d_i$$

时间复杂度 $O(Tnr)$ 。

博物馆

给一张 n 个点 m 条边的无向图，有两个人，每个人每分钟都可以：

1. 在第 i 个房间有 p_i 的概率不动；
2. 在第 i 个房间有剩下 $(1 - p_i)$ 的概率等概率地移动到有边连接的点。

一开始两个人在 a, b 两个房间，求他们在每个房间相遇的概率。

$$1 \leq n \leq 20, 1 \leq m \leq \frac{n(n-1)}{2}$$

BZOJ 3270 博物馆

套路消元题目。令 $f(i, j)$ 表示一个人到了 i ，一个人到了 j 的概率。

假设现在在 (i, j) 状态上，那么：（ d 表示度数）

1. 都不走： $f(i, j)p_i p_j$

2. i 是从 x 过来的， j 是留在这的： $f(x, j) \frac{1-p_x}{d_x} p_j$

3. j 是从 y 过来的， i 是留在这的： $f(i, y) p_i \frac{1-p_y}{d_y}$

4. i 是从 x 过来的， j 是从 y 过来的： $f(x, y) \frac{(1-p_x)(1-p_y)}{d_x d_y}$

从而：
$$f(i, j) = f(i, j)p_i p_j + f(x, j) \frac{1-p_x}{d_x} p_j + f(i, y) p_i \frac{1-p_y}{d_y} + f(x, y) \frac{(1-p_x)(1-p_y)}{d_x d_y}$$

BZOJ 3270 博物馆

对于每一个点都有一个方程，那么总共有 n^2 个方程。

接着对 n^2 个方程和未知数做高斯消元即可得到答案。

注意对于起点情况的特判。

时间复杂度 $O(n^6)$ 。

一种典型套路。

游走

一个无向连通图，顶点从1编号到 n ，边从1编号到 m 。

小Z在该图上进行随机游走，初始时小Z在1号顶点。

每一步小Z以相等的概率随机选择当前顶点的某条边。

沿着这条边走到下一个顶点，获得等于这条边的编号的分数。

当小Z到达 n 号顶点时游走结束，总分为所有获得的分数之和。

现在，请你对这 m 条边进行编号，使得小Z获得的总分的期望值最小。

$$2 \leq n \leq 500$$

BZOJ 3143 HNOI 2013 游走

如果我们知道每条边被走过的期望次数，就可以定下来权值得到答案了。

如果我们知道每个点经过的期望次数，我们就能求出每条边经过期望次数。

每条边经过期望次数=两个端点经过期望次数/两个端点度数。

BZOJ 3143 HNOI 2013 游走

如果我们知道每条边被走过的期望次数，就可以定下来权值得到答案了。

如果我们知道每个点经过的期望次数，我们就能求出每条边经过期望次数。

每条边经过期望次数=两个端点经过期望次数/两个端点度数。

而每个点经过的期望次数，就是一个高斯消元模型了。

$$f(i) = \sum_{(i,j)} \frac{f(j)}{d_j}$$

其中， d_j 为 j 度数。

进行高斯消元即可。时间复杂度 $O(n^3)$

类型总结

高斯消元类型：

列出方程后，方程形成了一个循环求解的过程，那么我们就可以利用高斯消元来进行求解
高斯消元的复杂度一般较高，所以相关题目一般数据范围可以支持至少 n^3 的级别。

One Person Game

有三个色子，第 i 个色子有 k_i 面，每面面值为 $1, 2, \dots, k_i$ ，每面出现的概率相同，均为 $\frac{1}{k_i}$ 。

每次扔出的面值加入总得分中。问期望扔多少次使得得分大于 n 。

为了给题目加大难度，我们规定：

当 $k_1 = a, k_2 = b, k_3 = c$ 时，得分清零。

多组数据。

$$1 \leq T \leq 300, 0 \leq n \leq 500, 1 < k_1, k_2, k_3 \leq 6$$

ZOJ 3329 One Person Game

设 $f(i)$ 表示达到 i 分时到达目标状态的期望次数。

p_k 表示扔出总点数为 k 的概率， p_0 表示扔出去后回到0分的概率（可以枚举预处理得到）。

那么：
$$f(i) = \sum p_k f(i + k) + p_0 f(0) + 1$$

高斯消元？

ZOJ 3329 One Person Game

设 $f(i)$ 表示达到 i 分时到达目标状态的期望次数。

p_k 表示扔出总点数为 k 的概率， p_0 表示扔出去后回到0分的概率（可以枚举预处理得到）。

那么： $f(i) = \sum p_k f(i+k) + p_0 f(0) + 1$

高斯消元？

我们设 $f(i) = A(i)f(0) + B(i)$

从而： $A(i)f(0) + B(i) = \sum_k p_k (A(i+k)f(0) + B(i+k)) + p_0 f(0) + 1$

即 $A(i)f(0) + B(i) = (p_0 + \sum_k p_k A(i+k))f(0) + (\sum_k p_k B(i+k) + 1)$

故： $A(i) = p_0 + \sum_k p_k A(i+k), B(i) = \sum_k p_k B(i+k) + 1$ 。

递推即可，时间复杂度 $O(n)$ 。

Maze

给定一棵 n 个点的树，小A初始在1号结点。

在 i 号结点被杀死的概率为 k_i ；走出迷宫的概率为 e_i 。

若被杀死，则返回1号节点重新出发。

若未走出迷宫且未被杀死，则等可能的走向与其相连的一个结点。

求期望走多少条边可以走出迷宫。

如果走不出迷宫，输出impossible。

多组数据。

$$1 \leq T \leq 30, 1 \leq n \leq 10^4$$

HDU 4035 Maze

设 f_x 表示当前从 x 号节点出发，走出迷宫的期望边数。

若 x 为叶子节点，有 $f_x = k_x \times f_1 + e_x \times 0 + (1 - k_x - e_x) \times (f_{par_x} + 1)$

若 x 为非叶子节点，有

$$f_x = k_x \times f_1 + e_x \times 0 + \frac{1}{m} (1 - k_x - e_x) \left(f_{par_x} + \sum_{y \text{ is a son of } x} f_y + m \right)$$

其中， m 为连出边数量。

高斯消元？

HDU 4035 Maze

设 f_x 表示当前从 x 号节点出发，走出迷宫的期望边数。

若 x 为叶子节点，有 $f_x = k_x \times f_1 + e_x \times 0 + (1 - k_x - e_x) \times (f_{par_x} + 1)$

若 x 为非叶子节点，有

$$f_x = k_x \times f_1 + e_x \times 0 + \frac{1}{m} (1 - k_x - e_x) \left(f_{par_x} + \sum_{y \text{ is a son of } x} f_y + m \right)$$

其中， m 为连出边数量，

我们发现，所有点的dp式子都可以写成：

$$f_x = a_x \times f_1 + b_x \times f_{par_x} + c_x$$

然后我们可以化简！

HDU 4035 Maze

若 x 为叶子节点, 有 $a_x = k_x, b_x = c_x = 1 - k_x - e_x$

若 x 为非叶子节点, 我们进一步化简:

$$\begin{aligned} f_x &= k_x f_1 + \frac{1}{m} (1 - k_x - e_x) \left(f_{par_x} + \sum_y (a_y f_1 + b_y f_x + c_y) + m \right) \\ f_x &= k_x f_1 + \frac{1}{m} (1 - k_x - e_x) \left(f_{par_x} + f_1 \sum_y a_y + f_x \sum_y b_y + \sum_y c_y + m \right) \\ &\quad \left(1 - \frac{1}{m} (1 - k_x - e_x) \sum_y b_y \right) f_x \\ &= \left(k_x + \frac{1}{m} (1 - k_x - e_x) \sum_y a_y \right) f_1 + \frac{1}{m} (1 - k_x - e_x) f_{par_x} + \frac{1}{m} (1 - k_x - e_x) \left(\sum_y c_y + m \right) \end{aligned}$$

HDU 4035 Maze

$$\begin{aligned} & \left(1 - \frac{1}{m}(1 - k_x - e_x) \sum_y b_y\right) f_x \\ &= \left(k_x + \frac{1}{m}(1 - k_x - e_x) \sum_y a_y\right) f_1 + \frac{1}{m}(1 - k_x - e_x) f_{par_x} + \frac{1}{m}(1 - k_x - e_x) \left(\sum_y c_y + m\right) \end{aligned}$$

故对应

$$\begin{aligned} a_x &= \frac{\left(k_x + \frac{1}{m}(1 - k_x - e_x) \sum_y a_y\right)}{\left(1 - \frac{1}{m}(1 - k_x - e_x) \sum_y b_y\right)}, b_x = \frac{\frac{1}{m}(1 - k_x - e_x)}{\left(1 - \frac{1}{m}(1 - k_x - e_x) \sum_y b_y\right)} \\ c_x &= \frac{\frac{1}{m}(1 - k_x - e_x)(\sum_y c_y + m)}{\left(1 - \frac{1}{m}(1 - k_x - e_x) \sum_y b_y\right)} \end{aligned}$$

HDU 4035 Maze

$$a_x = \frac{\left(k_x + \frac{1}{m}(1 - k_x - e_x) \sum_y a_y\right)}{\left(1 - \frac{1}{m}(1 - k_x - e_x) \sum_y b_y\right)}, b_x = \frac{\frac{1}{m}(1 - k_x - e_x)}{\left(1 - \frac{1}{m}(1 - k_x - e_x) \sum_y b_y\right)}$$
$$c_x = \frac{\frac{1}{m}(1 - k_x - e_x)(\sum_y c_y + m)}{\left(1 - \frac{1}{m}(1 - k_x - e_x) \sum_y b_y\right)}$$

一遍树形dp求出a、b、c数组。

对于1号节点，有 $f_1 = a_1 f_1 + b_1 \times 0 + c_1$ ，即 $\text{ans} = f_1 = \frac{c_1}{1 - a_1}$ 。

无解的情况即 $a_1 \rightarrow 1$ 。

复杂度 $O(n)$

类型总结

解系数类型：

列出方程后，方程形成了一个循环求解的过程，那么我们就可以利用高斯消元来进行求解。有时，数据范围不支持我们使用高斯消元，我们可以考虑解出所对应的系数。

解系数类型一般有一个特点，就是大多和一个特定的未知量相关。

如Maze题中与 f_1, f_{fa} 相关，而One Person Game中与 $f(0)$ 相关。

假面

记 C 为结界技能的数量。

$$n \leq 200, Q \leq 200000, C \leq 1000$$

$$m_i \leq 100$$

TL: 6s

针针喜欢玩一款叫做 DotA (Defense of the Algorithm) 的游戏, 在这个游戏中, 针针会操纵自己的英雄与队友一起对抗另一支队伍。

针针在 DotA 中最喜欢使用的英雄叫做假面 (Faceless), 该英雄有 2 个技能:

- 锁定: 对一名指定的敌方单位使用, 以 p 的概率对该单位造成 1 点伤害 (使其减少 1 点生命值)。
- 结界: 在一片区域施放结界, 让该区域内的所有其他单位无法动弹。

在游戏中, 如果一个单位的生命值降至 0 或 0 以下, 那么该单位就会死亡。

针针操纵假面的水平一般, 因此他决定勤加练习。

现在有 n 个敌方单位 (编号从 1 至 n), 编号为 i 的敌方单位有 h_i 点生命值。

针针已经安排好了练习的计划, 他会按顺序施放 Q 个技能:

- 对于锁定技能: 针针会指定一个敌方单位 id , 并对它施放。由于决定概率系数 p 的因素很多, 因此每次的 p 都不一定相同。
 - 特别地, 如果该敌方单位已经死亡, 那么该技能不会造成任何效果。
- 对于结界技能: 针针会希望对 k 个指定的敌方单位施放, 但由于针针并不擅长施放该技能, 因此他只能命中恰好 1 个敌方单位。命中每个存活的敌方单位的概率是相等的 (也就是说已经死亡的敌方单位不会有任何影响)。
 - 特别地, 如果这 k 个敌方单位均已死亡, 那么该技能同样不会命中任何敌方单位。

现在, 围观针针进行练习的绿绿想知道:

1. 对于针针施放的每个结界技能, 它命中各敌人的概率分别是多少。
2. 在针针的所有技能施放完毕后, 所有敌方单位剩余生命值的期望分别是多少。

由于绿绿还要围观针针训练, 所以请你帮他解决这两个问题。

为了防止精度误差, 对于所有需要输出的数值, 请输出其在模 998,244,353 意义下的值。

CTSC 2018 假面

考虑先求出每个人剩下的血的期望 $E_u = \sum_{i=0}^{m_i} ip(u, i)$

其中, $p^{(')} (u, i)$ 表示第 u 个人还剩 i 滴血的期望, 那么每次使用锁定的时候 $O(m)$ 转移即可。

$p'(u, i) = p \times p(u, i + 1) + (1 - p) \times p(u, i)$, 特别地, $p'(u, 0) = p(u, 0) + p \times p(u, 1)$

那么第一种操作就 $O(Qm)$ 做完了, 考虑结界操作。

设每个人活下来的概率为 $q(u) = 1 - p(u, 0)$ 。

k 个人中的每个人来说, 选中 u 的概率:

$$q(u) \sum_{i=0}^{k-1} \frac{f(u, i)}{i + 1}$$

其中 $f(u, i)$ 表示除了 u 之外只有 i 个人活着的概率。

CTSC 2018 假面

那么，要怎么求 $f(u, i)$ 呢？枚举另外一个人 v 来更新 $f(u, i)$ 得 $f'(u, i)$

$$f'(u, i) = q(v)f(u, i - 1) + (1 - q(v))f(u, i)$$

那这样时间复杂度为 $O(Qm + Cn^3)$

如何优化？

CTSC 2018 假面

那么，要怎么求 $f(u, i)$ 呢？枚举另外一个人 v 来更新 $f(u, i)$ 得 $f'(u, i)$

$$f'(u, i) = q(v)f(u, i - 1) + (1 - q(v))f(u, i)$$

那这样时间复杂度为 $O(Qm + Cn^3)$

考虑另外设 $g(i)$ 表示还活着任意 i 个人的概率。

观察到上述式子可以改写：

$$g(i) = (1 - q(u))f(u, i) + q(u)f(u, i - 1)$$

从而：

$$f(u, i) = \frac{g(i) - q(u)f(u, i - 1)}{1 - q(u)} \quad (q(u) \neq 1)$$

CTSC 2018 假面

当 $q(u) = 1$ 时, $f(u, i) = g(i + 1)$

$g(i)$ 可以通过一个 $O(n^2)$ 的dp求出:

$$g(i) = q(u)g(i - 1) + (1 - q(u))g(i)$$

经过这样处理后, f, g 都可以在 $O(n^2)$ 时间内求出

总复杂度: $O(Qm + Cn^2)$, 可以通过 本题。

非诚勿扰

有 n 个女性和 m 个男性要配对。

每个女性有一个列表，为这 m 个男性的子集（可能空）。

速配算法：将列表的男性按编号从小到大呈现，每次会有 P 的概率接受， $1 - P$ 的概率拒绝。

若拒绝，显示下一个男性，以此类推，若是最后一个则从头重新循环此过程。

最终，每个女性只能选择接受一个男性，而一个男性可能被许多女性接受。

考虑任意两个不同的、列表非空的女性 a, b

若 a 的编号小于 b 但是 a 选择的男性编号大于 b ，则称其为一对**不稳定因素**。

求不稳定因素期望对数。

$$1 \leq n, m \leq 5 \times 10^5, 0.4 \leq P \leq 0.6$$

BZOJ 4481 JSOI 2013 非诚勿扰

设 $P(x, y)$ 表示第 x 个女生抽中列表第 y 个男生的概率，则

$$P(x, y) = \sum_{k=0}^{\infty} (1-p)^{y-1+km} p = p \sum_{k=0}^{\infty} (1-p)^{y-1+km} = p \frac{(1-p)^{y-1}}{1-(1-p)^m}$$

知道概率后只要用类似树状数组求逆序对的方法即可，每次插入一个概率值统计即可。

时间复杂度 $O(n \log n)$

New Year and Arbitrary Arrangement

从一个空串开始，每次在串的末尾添加'a'或'b'。

其中，添加'a'的概率为 $\frac{pa}{pa+pb}$ ，添加'b'的概率为 $\frac{pb}{pa+pb}$ 。

当整个串中产生了 k 个及以上的'ab'子序列（可不连续）操作就会停止。

求在操作停止时序列中'ab'子序列的个数的期望。

$$1 \leq k \leq 10^3, 1 \leq pa, pb \leq 10^6$$

CodeForces 908D New Year and Arbitrary Arrangement

$f(i, j)$ 表示前缀包含 i 个'a'， j 个'ab'子序列的所有字符串，期望得到的'ab'子序列个数。

$$f(i, j) = f(i + 1, j) \times \frac{pa}{pa+pb} + f(i, j + 1) \times \frac{pb}{pa+pb} \quad (\text{枚举添加'a'或'b'})$$

初始状态为 $f(i, j) = j (j \geq k)$

目标状态为 $f(0, 0)$ 。

但是由于存在无穷的情况，无法转移。

CodeForces 908D New Year and Arbitrary Arrangement

$$f(i, j) = f(i + 1, j) \times \frac{pa}{pa+pb} + f(i, j + 1) \times \frac{pb}{pa+pb} \quad (\text{枚举添加'a'或'b'})$$

初始状态为 $f(i, j) = j (j \geq k)$

目标状态为 $f(0, 0)$ 。

考虑 $f(i, j) (i + j \geq k)$ 的情况，只要后面出现一个 **b**，立刻就结束，那么：

$$\text{令 } P = \frac{pb}{pa+pb}, \text{ 则 } f(i, j) = i + j + (1 - P)P + 2(1 - P)^2P + 3(1 - P)^3P + \dots = i + j + \frac{pa}{pb}。$$

从而即可转移。时间复杂度 $O(k^2)$ 。

收集邮票

有 n 种不同的邮票，皮皮想收集所有种类的邮票。

每次只能买一张，并且买到的邮票是 n 种的哪一种是等概率的，概率均为 $\frac{1}{n}$ 。

皮皮购买第 k 张邮票需要支付 k 元钱。

现在皮皮手中没有邮票，皮皮想知道自己得到所有种类的邮票需要花费的钱数目的期望。

$$1 \leq n \leq 10^4$$

BZOJ 1426 收集邮票

令 $g(i)$ 表示现在有 i 张，要买到 n 张的期望次数， $P(x, i)$ 为买 x 次能从 i 种买到 n 种的概率。

那么：

$$g(i) = \sum_{x=0}^{\infty} xP(x, i)$$

那么，又有：

$$g(i) = g(i + 1) + \frac{n}{n - i}$$

BZOJ 1426 收集邮票

设 $f(i, j)$ 表示现在有 i 张，下一张 j 元，买到 n 张的期望，答案显然是 $f(0, 1)$ 。

所以转移显然：

$$f(i, j) = \frac{i}{n} \times f(i, j + 1) + \frac{n - i}{n} f(i + 1, j + 1) + j$$

但是 j 这一维可以达到 $+\infty$ ，不能直接地推，考虑：

$$f(i, j) = \sum_{x=0}^{\infty} (j + (j + 1) + (j + 2) + \cdots + (j + x - 1)) P(x, i)$$
$$f(i, j) = \sum_{x=0}^{\infty} \frac{x}{2} (x + 2j - 1) P(x, i)$$

BZOJ 1426 收集邮票

$$f(i, j) = \sum_{x=0}^{\infty} \frac{x}{2} (x + 2j - 1) P(x, i)$$

作差：

$$f(i, j + 1) - f(i, j) = \sum_{x=0}^{\infty} x P(x, i) = g(i)$$

带入下式

$$f(i, j) = \frac{i}{n} \times f(i, j + 1) + \frac{n - i}{n} f(i + 1, j + 1) + j$$

有

$$f(i, j) = \frac{i}{n} (f(i, j) + g(i)) + \frac{n - i}{n} (f(i + 1, j) + g(i + 1)) + j$$

BZOJ 1426 收集邮票

$$f(i, j) = \left(\frac{n-i}{n} \times (f(i+1, j) + g(i+1)) + \frac{i}{n} \times g(i) + j \right) \times \frac{n}{n-i}$$

观察式子会发现：神奇的事情出现了！

这个递推式对于一个确定的 j ，在 j 这一维就不涉及到除 j 以外的其他值了。

由于我们求 $f(0,1)$ ，令 $j=1$ 代入：

$$f(i, 1) = \left(\frac{n-i}{n} \times (f(i+1, 1) + g(i+1)) + \frac{i}{n} \times g(i) + 1 \right) \times \frac{n}{n-i}$$

即得答案。时间复杂度 $O(n)$ 。

Puzzles

有 n 个节点的树，你从根开始随便走。

每当你第一次到达一个点时，把这个点的权记为（你已经经过的不同的点数+1）。

每一次只有当子树中所有的点都游走完才会向父亲走，走到每个儿子的概率相同。

对于每个点，求权的期望。

$$1 \leq n \leq 10^5$$

CodeForces 696B Puzzles

首先，所有子树中的节点的权一定比父亲大。

而且子树中的大小关系和我们的游走顺序有关。

对于一个节点 u ，设他有 k 个儿子： v_1, v_2, \dots, v_k ，那么访问顺序构成了一个全排列！

那么对于任意一个节点 v_i ，我们可以发现，在全排列中， v_j 在它前面的概率均为0.5！

所以， $E(v_i) = E(u) + 1 + \frac{1}{2} \sum_j sz_{v_j} = E(u) + 1 + \frac{1}{2} (sz_u - 1 - sz_{v_i})$ 。

一次树形dp即可求出答案，时间复杂度 $O(n)$ 。

概率充电器

有 n 个结点，相互之间通过 $n - 1$ 条边连接，形成一棵树。

第 i 个节点有 a_i 的概率直接充电，第 i 条边有 b_i 的概率导电。

求期望有电的结点数。

$$1 \leq n \leq 10^6$$

BZOJ3566 概率充电器

设计状态：

f_x 表示不管父亲及祖先， x 充上电的概率。

g_x 表示 x 点父亲向 x 点导电的概率。

这样设计状态在转移的时候需要用到概率加法，比较麻烦。

考虑换一种状态设计？

BZOJ3566 概率充电器

设计状态：

f_x 表示不管父亲及祖先， x 充上电的概率。

g_x 表示 x 点父亲向 x 点导电的概率。

这样设计状态在转移的时候需要用到概率加法，比较麻烦。

考虑换一种状态设计？直接反面设计即可。

f_x 表示不管父亲及祖先， x 充~~不~~上电的概率。

g_x 表示 x 点父亲~~不~~向 x 点导电的概率。

BZOJ3566 概率充电器

f_x 表示不管父亲及祖先， x 充不上电的概率。

g_x 表示 x 点父亲不向 x 点导电的概率。

f_x 转移比较简单， $f_x = (1 - a_x) \times \prod (f_{son} + (1 - f_{son}) \times (1 - b_{son \rightarrow x}))$

先用一遍dfs把 f_x 处理出来，然后用类似up-down dp的方法把父亲信息传递下去，得出 g_x 。

父亲有没有电可以通过 x 节点和 x 节点的爷爷传递，也可以通过 x 节点的兄弟传递。通过 x 节点兄弟或 x 节点的爷爷传递，仍然没有电的概率：

$$t = g_{fa} \times f_{fa} \div (f_x + (1 - f_x) \times (1 - b_{fa \rightarrow x}))$$

那么 $g_x = t + (1 - t) \times (1 - b_{fa \rightarrow x})$ 。

最后每个节点有电概率就是 $1 - f_x \times g_x$ 。

时间复杂度 $\Theta(n)$

Maze

一个迷宫是一棵 n 个点的树。迷宫的起点和终点都按照某种概率随机选取。

人们会在迷宫中用DFS寻找终点，如果有很多条路径，会等概率选择一条。

考虑如下伪代码：

```
DFS(x)
  if x == exit vertex then
    finish search
  flag[x] <- TRUE
  random shuffle the vertices' order in V(x) // here all permutations have equal probability to be chosen
  for i <- 1 to length[V] do
    if flag[V[i]] = FALSE then
      count++;
      DFS(y);
  count++;
```


Maze

```
DFS(x)
  if x == exit vertex then
    finish search
  flag[x] <- TRUE
  random shuffle the vertices' order in V(x) // here all permutations have equal probability to be chosen
  for i <- 1 to length[V] do
    if flag[V[i]] = FALSE then
      count++;
      DFS(y);
  count++;
```

其中， $V(x)$ 是与 x 相连的顶点列表，最初flag数组均为0。

从起点开始调用DFS，当搜索终止，count的值就是所走步数。

对于每个点给出两个值 x_i, y_i ，那么这个点作为起点概率为 $\frac{x_i}{\sum_{j=1}^n x_j}$ ，作为终点概率 $\frac{y_i}{\sum_{j=1}^n y_j}$ 。

求迷宫从入口走到出口期望步数。 $1 \leq n \leq 10^5$

CodeForces 123E Maze

考虑固定起点和终点，那么对于所有可能被经过的边，只有两种情况：

1. 在起点到终点的路径上，必须被经过1次，且只能被经过1次；
2. 不在起点到终点的路径上，有可能被经过0次或2次。

CodeForces 123E Maze

考虑固定起点和终点，那么对于所有可能被经过的边，只有两种情况：

1. 在起点到终点的路径上，必须被经过1次，且只能被经过1次；
2. 不在起点到终点的路径上，有可能被经过0次或2次。

考虑对于当前点的每个子树，终点不在的子树进入的概率为 $\frac{1}{2}$ 。为什么？

CodeForces 123E Maze

考虑固定起点和终点，那么对于所有可能被经过的边，只有两种情况：

1. 在起点到终点的路径上，必须被经过1次，且只能被经过1次；
2. 不在起点到终点的路径上，有可能被经过0次或2次。

考虑对于当前点的每个子树，终点不在的子树进入的概率为 $\frac{1}{2}$ 。为什么？

除起点所在子树外，设有 k 个子树，那么也对应着一个全排列！

设终点所在子树为 i ，对于任意其他子树 j ，一定有一半的全排列满足 i 在 j 之后访问。

由于一一对应关系，所以每个子树访问的概率为 $\frac{1}{2}$ ！

CodeForces 123E Maze

所以，期望次数 $(0 + 2) \times \frac{1}{2} = 1$ ，也就是说：

所有可能被访问的边，期望访问次数均为1次。

如果 e 是终点，那么：

1. 当起点在 e 子树内时，从子树向上到达 e 就会停止，所有可能经过的边就是子树内所有边。
2. 当起点不在 e 子树内时，所有可能经过的边就是除 e 子树外所有边。

一遍树形dp就可以 $O(n)$ 解决。

图样

西方有 n 个国家，长者决定向西方的每个国家普及人生经验，但首先要让他们互通火车。

第 i 个国家有权值 a_i ，修建连接第 i 个国家到第 j 个国家的铁路，需要付出 $a_i \text{ xor } a_j$ 的代价。

长者希望代价总和尽量小（也就是选择一个最小生成树）。

我们只知道每个国家的权值都是一个在 0 到 $2^m - 1$ 之间的随机整数。

长者希望知道他所需要付出的代价的期望。

对质数取模。

$$1 \leq n \leq 50, 1 \leq m \leq 8$$

BZOJ 4770 图样

设 $f(i, j)$ 表示 i 个点，点权为 $[0, 2^j - 1]$ 的随机值时所有情况的最小生成树之和。

那么答案等于 $f(n, m)$ 除以 2^{nm} 。

考虑如何求最小生成树，根据第 m 为1还是0可以将点分成 S 和 T 两个集合。

那么 S 和 T 内部一定形成生成树，再找一条连接 S, T ，代价最小的边即可。

用 $g(i, j, k)$ 表示一边有 i 个点，一边有 j 个点，点权为 $[0, 2^k - 1]$ 的随机值时所有情况下的最小边的和。那么，

BZOJ 4770 图样

$$f(i, j) = \sum_{k=0}^i \binom{i}{k} \left((2^{j-1})^{i-k} f(k, j-1) + (2^{j-1})^k f(i-k, j-1) + g(k, i-k, j-1) + 2^{j-1} \times (2^{j-1})^i \right)$$

意义：从 i 个点选出 k 个点划分到 S 集合，那么剩下 $i-k$ 个点在 T 集合。

S 集合内部的生成树和为 $f(k, j-1)$ ，这时候， T 集合可以随便取权值，共有 $(2^{j-1})^{i-k}$ 种情况。

同理， T 集合内部生成树和为 $f(i-k, j-1)$ ， S 集合随便取权值，共有 $(2^{j-1})^k$ 种情况。

考虑相连的边，答案显然为 $g(k, i-k, j-1)$ （不考虑最高位）

最高位只能必须出来一个1，所以所有 $(2^{j-1})^i$ 种方案都有贡献 2^{j-1} （ S 到 T 的边贡献的）。

BZOJ 4770 图样

考虑 $g(i, j, k)$ 。我们设 $p(i, j, k, l)$ 表示左边 i 个点右边 j 个点，点权 $[0, 2^k - 1]$ 随机，两部分之间最小边权大于等于 l 方案数。

那么， $g(i, j, k) = \sum_{l=1}^{2^k-1} 2(p(i, j, k, l) - p(i, j, k, l+1)) \times l + p(i, j, k, 2^k - 1) \times (2^k - 1)$

化简后，

$$g(i, j, k) = \sum_{l=1}^{2^k-1} p(i, j, k, l)$$

然后呢？怎么求 p ？

BZOJ 4770 图样

$p(i, j, k, l)$ 表示左边 i 个点右边 j 个点，点权 $[0, 2^k - 1]$ 随机，两部分之间最小边权大于等于 l 的方案数。

$$p(i, j, k, l) = \sum_{x=0}^i \sum_{y=0}^j \binom{i}{x} \binom{j}{y} p(x, y, k-1, l) p(i-x, j-y, k-1, l)$$

枚举左边和右边各有多少个最高位为1的。那么最小边一定是两个端点最高位一样。

边界： $p(i, j, 1, 1) = 2, p(i, j, k, 0) = (2^k)^{i+j}$ 。记忆化搜索即可。

时间复杂度 $O(n^4 2^m)$ 。

~~这种题当然是写完正解打表啊.....~~

习题

统一提交地址: <https://vjudge.net/contest/283105>

密码: hailiangC0212