浅析竞赛中一类数学期望问题的解决方法

福建省福州第八中学　汤可因

摘要

期望在我们生活中有着十分广泛的应用，而对于我们信息学奥赛也出现了不少求解期望值的问题。本文对于竞赛中涉及求离散型随机变量的数学期望的题目所使用的常用方法归纳成为两大类，并进行了总结与分析。

关键字

期望 递推 动态规划 方程组

目录

引言

数学期望亦称为期望，期望值等，在概率论和统计学中，一个离散型随机变量的期望值是试验中每次可能结果的概率乘以其结果的总和。

而期望在我们生活中有着十分广泛的应用。例如要设计一个彩票或赌博游戏，目标为赢利，那么计算能得到的钱以及需要付出的钱的期望，它们的差则需要大于0。又例如对于是否进行一项投资的决策，可以通过分析总结得出可能的结果并估算出其概率，得到一个期望值而决定是否进行。期望也许与每一个结果都不相等，但是却是我们评估一个事情好坏的一种直观的表达。

正因为期望在生活中有如此之多的应用，对于我们信息学奥赛也出现了不少求解期望值的问题。而其中大多数又均为求离散型随机变量的数学期望。本文对于这类题目所会涉及到的常用方法进行了归纳，总结与分析。

预备知识

一、期望的数学定义

如果 X 是一个离散的随机变量，输出值为 $x_1, x_2, \ldots$， 和输出值相应的概率为 $p_1, p_2, \ldots$ （概率和为 1），那么期望值 $E(X)=\sum_i p_i x_i$ 。

例如投掷一枚骰子，X 表示掷出的点数，$P(X=1)$，$P(X=2)$，…，$P(X=6)$均为 $\frac{1}{6}$，那么 $E(X)=1\cdot\frac{1}{6}+2\cdot\frac{1}{6}+3\cdot\frac{1}{6}+4\cdot\frac{1}{6}+5\cdot\frac{1}{6}+6\cdot\frac{1}{6}=\frac{1+2+3+4+5+6}{6}=3.5$ 。

二、期望的线性性质

对于任意随机变量 X 和 Y 以及常量 a 和 b，有

$E(aX+bY)=aE(X)+bE(Y)$

当两个随机变量 X 和 Y 独立且各自都有一个已定义的期望时，有

$E(XY)=E(X)E(Y)$

三、全概率公式

假设{ $B_n \mid n = 1, 2, 3, \ldots$ } 是一个概率空间的有限或者可数无限的分割，且每个集合 B_n 是一个可测集合，则对任意事件 A 有全概率公式：

$$P(A) = \sum_n P(A \mid B_n) P(B_n)$$

其中 $P(A \mid B)$是 B 发生后 A 的条件概率

四、条件期望与全期望公式

$$p_{ij} = P(X = x_i，\ Y = y_j) \quad (i，\ j = 1,2,\cdots)$$

当 $X=x_i$ 时，随机变量 Y 的条件数学期望以 $E(Y \mid X=x_i)$表示。

全期望公式：

$$\begin{aligned} E(E(Y \mid X)) &= \sum_i P(X = x_i) E(Y \mid X = x_i) \\ &= \sum_i p_i \sum_k y_k \frac{p_{ik}}{p_i} \\ &= \sum_i \sum_k p_i y_k \frac{p_{ik}}{p_i} \\ &= \sum_i \sum_k y_k p_{ik} \\ &= E(Y) \end{aligned}$$

所以 $E(Y) = E(E(Y \mid X)) = \sum_i P(X = x_i) E(Y \mid X = x_i)$

例如，一项工作由甲一个人完成，平均需要 4 小时，而乙有 0.4 的概率来帮忙，两个人完成平均只需要 3 小时。若用 X 表示完成这项工作的人数，而 Y 表示完成的这项工作的期望时间（单位小时），由于这项工作要么由一个人完成，要么由两个人完成，那么这项工作完成的期望时间 $E(Y) = P(X = 1)E(Y \mid X = 1) + P(X = 2)E(Y \mid X = 2) = (1－0.4)\times 4－0.4\times 3 = 3.6$。

一、利用递推或动态规划解决

递推作为一种基础的算法，相信所有人都不会陌生。而对于一部分期望问题而言，递推则为一种快速有效的方法。我们不需要将所有可能的情况都枚举出来，而是根据已经求出的期望推出其它状态的期望，亦或根据一些特点和结果相同的情况，求出其概率。

例题一：聪聪与可可[1]

在一个魔法森林里，住着一只聪明的小猫聪聪和一只可爱的小老鼠可可。

整个森林可以认为是一个无向图，图中有 N 个美丽的景点，景点从 1 至 N 编号。在景点之间有一些路连接。

可可正在景点 M ($M \leqslant N$)处。以后的每个时间单位，可可都会选择去相邻的景点（可能有多个）中的一个或停留在原景点不动。而去这些地方所发生的概率是相等的。

聪聪是很聪明的，所以，当她在景点 C 时，她会选一个更靠近可可的景点，如果这样的景点有多个，她会选一个标号最小的景点。如果走完第一步以后仍然没吃到可可，她还可以在本段时间内再向可可走近一步。

在每个时间单位，假设聪聪先走，可可后走。在某一时刻，若聪聪和可可位于同一个景点，则可可就被吃掉了。

问平均情况下，聪聪几步就可能吃到可可。

分析：

根据题目要求聪聪会向可可不断靠近，且边无权，所以先进行 N 次广度优先搜索，预处理出 $p[i, j]$表示顶点 i 到顶点 j 的最短路上与顶点 i 相邻且编号最小的顶点编号。即聪聪在景点 i，可可在景点 j 时，聪聪第 1 步会走到的景点编号。

设 $f[i, j]$来表示聪聪在顶点 i，可可在顶点 j 时聪聪抓住可可的平均步数。令 $w[i, j]$表示与顶点 i 相邻的 j 个点编号，而用 $t[i]$表示顶点 i 的度。

---

[1] 题目来源：NOI 2005

可以确定聪聪下一步所在的顶点即 $p[p[i,j],j]$，可可下一步在顶点 $w[i,j]$，概率为 $\frac{1}{t[i]+1}$，下一步这个情况下的期望 $f[p[p[i,j],j],w[i,j]]$ 已经计算出，那么就是比 $f[p[p[i,j],j],w[i,j]]$ 多出一步。可可在原地停留的情况则类似。

所以可以得到：

$$f[i,j]=\frac{\sum_{k=1}^{t[i]} f[p[p[i,j],j],w[j,k]]+f[p[p[i,j],j],j]}{t[i]+1}+1$$

当然，$f[i,i]=0$，因为聪聪和可可已经在同一个点。若 $p[p[i,j],j]=j$ 或 $p[i,j]=j$ 则说明在这一步聪聪即可吃掉可可，那么 $f[i,j]=1$。

用记忆化搜索即可解决。

小结：

对于一些题目，可以期望间的递推关系，而其也可以认为是对应了全概率公式和期望的性质的应用。这是比较常见的也是最直观的一种状态设计方法和解法。对于第二部分所要提到的利用方程组解，也是针对“循环”的情况而基于这样的一种解法得来。

例题二：**Highlander**[2]

随机给出 1，2，…，n 个数字的一个错位排列 $i_1i_2\ldots i_n$（也就是等概率选择 n 个数字所有错位排列中的一个），对应了一张有向图 $G=(V,E)$，其中 $V=\{1,2,\ldots,n\}$，$E=\{(1,i_1),(2,i_2),\ldots,(n,i_n)\}$。问在最长环上的顶点数的期望值。

$2\le n\le 100$。

分析：

根据题目描述，则最后得到的有向图所有顶点出度入度均为 1，所以有向图

[2] 题目来源：SGU 385

必然由若干个不相交的环组成。而且由于是错位排列，所以没有自环。那么，我们通过不断确定排列中若干个数字，使其在同一个环内。为了避免重复，环的长度为从小到大确定。如果排列中剩下 m 个数字没有确定，并要使其中 k 个数字属于同一个长度为 k 的环，那么有 $\frac{A_m^k}{k}$ 种方案。如果有 i 个环长度相同，则还需要除以 i 的全排列也就是 A_i^i 。

那么用 $f[i, j, k]$ 表示了错位排列中 i 个数字已经确定，最长环为 j，且有 k 个长度为 j 的环的方案数。例如当 $n=4$ 时，$f[2, 2, 1]$ 有 $\frac{A_4^2}{2}=6$ 种方案：21**，3*1*，4*1*，*32*，*4*2，**43，$f[4, 2, 2]$ 可以由 $f[2, 2, 1]$ 得来，由于有 2 个重复的环则还需要除以全排列所以 $f[4,2,2]=\frac{f[2,2,1]\times\frac{A_2^2}{2}}{2}=3$，而对应了 2143，3412，4321 这 3 种错位排列方案。

状态转移方程就可以得出：

$$f[i,j,k]=\begin{cases}\dfrac{A_n^i}{i} & (i=j,\ i>1,\ k=1)\\ \dfrac{f[i-j,j,k-1]*\dfrac{A_{n-i+j}^j}{j}}{k} & (2\le j\le i-2, 1<k<\left\lfloor\dfrac{i}{j}\right\rfloor)\\ \sum_{l=2}^{\min(j,i-j+1)} g[i-j,l]*\dfrac{A_{n-i+j}^j}{j} & (2\le j\le i-2,\ k=1)\\ 0 & (\text{其他情况})\end{cases}$$

$$g[i,j]=\sum_{k=1}^{\left\lfloor\frac{i}{j}\right\rfloor} f[i,j,k]$$

答案即为 $\dfrac{\sum_{j=2}^{j\le n}\sum_{k=1}^{k\le n} j*k*f[n,j,k]}{\sum_{j=2}^{j\le n}\sum_{k=1}^{k\le n} f[n,j,k]}$

虽然说 $f[i, j, k]$ 可能很大，但是由于只是用于计算概率，用实数类型记录依然可以满足题目要求

注：

1、答案中分母的$\sum_{j=2}^{j\le n}\sum_{k=1}^{k\le n} f[n,j,k]$即为 n 个数字错位排列的排列总数。若用 $d[n]$ 表示 n 个数字的错排方案数，那么有递推式 $d[n] = (n - 1)(d[n - 1] + d[n - 2])$，其中 $d[1] = 0$，$d[2] = 1$。

2、可以用二维的状态得到一个更好的复杂度，但是上述给出的是最直观的一种方法。

小结：

对于一些问题比较难找到期望的递推关系，但是枚举对于多数题目依旧不现实的情况下。这时可以利用期望的定义即 $E(X) = \sum_i p_i x_i$ 。一般来说，对于这类问题，我们可以根据实际情况以概率或方案数（除以总方案数依旧等于概率）作为状态，而下标直接或间接对应了这个概率下的变量值。而问题就变成比较一般的统计方案或者利用全概率公式计算概率的递推问题。

例题三：**RedIsGood**[3]

桌面上有 R 张红牌和 B 张黑牌，随机打乱顺序后放在桌面上，开始一张一张地翻牌，翻到红牌得到 1 美元，黑牌则付出 1 美元。可以随时停止翻牌，在最优策略下平均能得到多少钱。

分析：

用 $f[i,j]$ 表示剩下 i 张红牌和 j 张黑牌获得钱的期望。决策只有 2 种，即翻牌与不翻牌。不翻牌期望为 0。翻到红、黑牌的概率分别为 $\frac{i}{i+j}$、$\frac{j}{i+j}$，而分别会

[3] 题目来源：TopCoder SRM420 Div1

得到、损失 1 美元，所以期望为 $(f[i-1,j]+1)*\frac{i}{i+j}+(f[i,j-1]-1)*\frac{j}{i+j}$。

那么可以比较容易地得出下面的转移方程：

$$f[i,j]=\begin{cases}\max\{0,(f[i-1,j]+1)*\frac{i}{i+j}+(f[i,j-1]-1)*\frac{j}{i+j}\} & i=0,j=0\\ f[i-1,j]+1 & i>0,j=0\\ 0 & i=0\end{cases}$$

那么 $f[R,B]$则为所求。

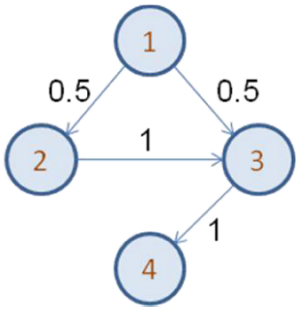
小结：

对于一些求期望并且带有决策的题需要用动态规划来解决，这类题由于要满足最优子结构，一般用期望来表示状态，而期望正表现了这个状态的优或劣。例如上题，若继续翻牌的期望小于 0 则说明翻牌平均情况下会损失更多钱，那么还不如就此收手。

二、建立线性方程组解决

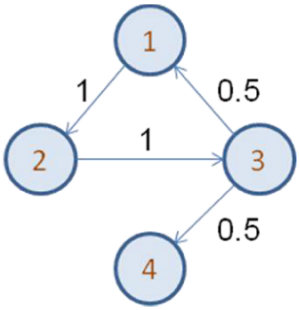
引入

给出一个有向图 $G=(V,E)$，点带权，顶点 i 的权值为 W_i。对于每条边 $(u,v)\in E$ 有一个属性 $P_{u,v}$，且 $P_{u,v}$ 为正数，其中 $P_{u,v}$ 表示从顶点 u 经过边(u,v)到顶点 v 的概率。若某点 i 发出边概率和为 P_i，即 $S_i=\sum_{(i,j)\in E}p_{i,j}$，那么在顶点 i 时有 $1-P_i$ 的概率停止行动。定义路径 $path=<v_1,v_2,\cdots>$的权为 $\sum_i W_{v_i}$，即这条路径上所有点权之和。问从一个顶点 s 开始，在每次按照指定的概率走的前提下，在某一顶点停止行动时所走的路径权的期望值。



例如右图，$s=1$，$W_1=W_2=W_3=1$，$W_4=0$。

可以看到从起点到停止行动有两条路径，这两条路径权分别为 3 和 2，而走这两条路径的概率均为 0.5。所以能得到期望值为 $3\times0.5+2\times0.5=2.5$。



若调整一条边的方向并修改相关概率，由于图中出现了环，这样路径就会有无数条。所以我们必须找到一种有效的方法处理一般的情况。

分析：

设 $F_{i,j}$ 为从顶点 i 出发，经过 j 步的权和的期望，那么可以得到：

$F_{i,0}=W_i$，当 $j>0$时，可以分成停止行动和走到某一个相邻点这两类情况。若存在一条边(i,k)，走这条边的概率 $P_{i,k}$，那么从顶点 i 出发经过顶点 k 的期望值即为顶点 k 出发 $j-1$ 步的期望 $F_{k,j-1}$ 加上顶点 i 的权值 W_i。当然，在点 i 时还有 $(1-\sum_{(i,k)\in E}P_{i,k})$ 的概率停止行动，而停止行动依然要计算顶点 i 的权值 W_i，所以有：

$$F_{i,j}=\sum_{(i,k)\in E}P_{i,k}(F_{k,j-1}+W_i)+(1-\sum_{(i,k)\in E}P_{i,k})W_i=\sum_{(i,k)\in E}P_{i,k}F_{k,j-1}+W_i$$

若 $F_{i,j}$ 当 $j\to\infty$ 时收敛，设收敛于 F_i，那么答案即为 F_s，可以通过迭代法求出满足精度要求的解。但是显然当精度要求较高或增长速度较慢，程序运行时间就将会很长，并且当极限不存在时也难以判断。

所以可以抛弃步数的思想，设 F_i 为从顶点 i 出发的权和期望，对于每个 F_i 类似地建立这样一个等式 $F_i=\sum_{(i,j)\in E}P_{i,j}F_j+W_i$，那么 F_s 则为所需要的答案。当图不存在环的情况下，那么我们就可以利用拓扑序进行递推以求得一个更好的复杂度。但是对于一般情况，显然按照这个式子递推就不完全可行了。这时利用所有的等式建立线性方程组求解则是成为了一个有效的方法。一般情况下，接下来即为利用高斯消元求解这个线性方程组。这样既得到了一个稳定的复杂度，根据方程组是否有唯一解的情况也能判断极限是否存在。

同一般线性方程组一样，这个方程依然面对着无解或无数组解的问题。而方程组没有唯一解并不代表所有未知数没有唯一解，例如下面这个方程组：

$$\begin{cases}a=1\\b=b+1\end{cases}$$

由于 a 与 b 无关所以 b 无解不影响 a 有唯一解。

由于每个顶点对应了方程组中的一个变量，那么方程组中只需要保留与顶点 s 有关，即 s 到达的顶点即可。实际操作中也可以对高斯消元的过程作一些相关调整。

而若需要求的未知数仍没有唯一解，一般情况下可以认为期望不存在。但是遇到具体问题时，需要根据题目的条件具体分析（例如下文会提到的例题 Mario）。

这可以作为绝大多数以期望作为状态的期望问题的模型，而对于无环的情况递推可以得到一个不错的时间复杂度，而对于一般情况高斯消元则成为通常解决这类期望问题的一把利器。

需要注意的地方：

- 若权在边上而不在点上的话，即边(u, v)的权值为 $W_{u,v}$，那么同理方程即为 $F_i = \sum_{(i,j)\in E} P_{i,j}(F_j + W_{i,j})$。
- 可以调整消元顺序让所要求的 E_s 放在最后，这样就可以不用回代，让实现更简洁，并在一些问题上能节省常数时间。

例题四：**First Knight**[4]

一个矩形区域被分成 $m*n$ 个单元编号为 $(1, 1)$至 (m, n)，左上为 $(1, 1)$，右下为(m, n)。给出 $P_{i,j}^{(k)}$，其中 $1 \le i \le m$，$1 \le j \le n$，$1 \le k \le 4$，表示了 (i, j)到 $(i+1, j)$，$(i, j+1)$，$(i-1, j)$，$(i, j-1)$的概率。一个骑士在 $(1, 1)$，按照给定概率走，每步都于之前无关，问到达 (m, n)的期望步数。

题目保证对于 $i \ne m$ 或 $j \ne n$，有 $\sum_{k=1}^{4} P_{i,j}^{(k)} = 1$，且 $P_{i,j}^{(1)}$ 和 $P_{i,j}^{(2)}$ 中至少一个不为 0。且保证走出矩形的概率与 $P_{m,n}^{(k)}$ 均为 0，答案不超过 1000000。

$1 \le m, n \le 40$。

分析：

初看这道题可以发现这题直接对应了上面给出的模型，而且题目条件 $P_{i,j}^{(1)}$ 和 $P_{i,j}^{(2)}$ 中至少一个不为 0，即所有点到达结束点(m, n)的概率均为 1，说明这个方程组不存在特殊情况，可以很容易地列出方程：

$E_{i,j} = P_{i,j}^{(1)} E_{i+1,j} + P_{i,j}^{(2)} E_{i,j+1} + P_{i,j}^{(3)} E_{i-1,j} + P_{i,j}^{(4)} E_{i,j-1} + 1$，其中 $1 \le i \le m$，$1 \le j \le n$。

但是由于未知数个数为 mn，那么时间复杂度则达到 $O(m^3 n^3)$，再加上数据组数非常多，虽然时限为 30s，直接套用高斯消元对于数据范围 $m, n \le 40$ 显然还是无法接受的。但是否说明这种方法不能用于此题呢？答案是否定的。

---

[4] 题目来源：SWERC 2008 Problem B

一、直观的优化。

1、 避免与 0 相乘。观察增广矩阵的特点，由于 $E_{i,j}$ 最多与 4 项有关，所以增广矩阵中会有很多的 0。对于两行消元时若某行的主元素所在列的元素为 0，那么显然这两行就不需要进行消元操作。而由于乘法的常数较大，所以消元时判断乘数都不为 0 时才进行乘法运算。

2、 调整消元顺序。让 $E_{1,1}$ 最后进行消元，这样可以不用回代。让满足 $(i+j) \bmod 2 = 1$（类似国际象棋棋盘其中的一色）的 $E_{i,j}$ 先进行消元。由于这些先进行消元的项互不相关，即消掉一项后另外几项的方程不会改变，所以每个方程最多与 4 个方程进行消元。实践证明，这么调整顺序会节省大约一半的时间。

进行以上两点优化之后即可通过此题，但是提交[5]后运行时间为 22*s*，不是很理想。

二、进一步思考。

虽然用上述优化本题在时限内得到解决，但是时空复杂度依然不很理想，为什么一个看似[6]时间复杂度依然为 $O(n^3m^3)$ 的算法依旧可以快速运行出极限数据的结果？有没有更好的方法？

观察方程：

$$E_{i,j} = P_{i,j}^{(1)}E_{i+1,j} + P_{i,j}^{(2)}E_{i,j+1} + P_{i,j}^{(3)}E_{i-1,j} + P_{i,j}^{(4)}E_{i,j-1} + 1$$

对于 $i = 1$，由于走出矩形的概率为 0，有 $P_{i,j}^{(3)} = 0$，

$$E_{1,j} = P_{1,j}^{(1)}E_{2,j} + P_{i,j}^{(2)}E_{1,j+1} + P_{1,j}^{(4)}E_{1,j-1} + 1$$

对于所有的 $E_{1,j}$，将 $E_{2,j}$ 看作常数项进行消元，对那么可以用只含有 $E_{2,1}$，$E_{2,2}$,…，$E_{2,n}$ 的代数式来表示 $E_{1,j}$，即：

$E_{1,j} = a_1E_{2,1} + a_2E_{2,2} + \cdots + a_nE_{2,n} + c_{1,j}$，其中 a_i，c 分别表示消元后方程的系

[5] 提交地址：http://acmicpc-live-archive.uva.es/nuevoportal/data/problem.php?p=4297

[6] 下文会说明事实上优化后的时间复杂度为 $O(n^3m^2)$，且常数小，所以能取得不错的效果。

数和常数。

将 $E_{1,j}$ 代入方程 $E_{2,j}=P_{2,j}^{(1)}E_{3,j}+P_{2,j}^{(2)}E_{2,j+1}+P_{2,j}^{(3)}E_{1,j}+P_{2,j}^{(4)}E_{2,j-1}+1$ 后，对于所有 $E_{2,j}$ 便可以用 $E_{2,1}$，$E_{2,2}$,…，$E_{2,n}$ 和 $E_{3,j}$ 表达出，同样将 $E_{3,j}$ 当作常数项进行消元。

同理，不断将 $E_{i,j}$ 以 $E_{i+1,k}$，$k=1,2,\cdots,m$ 表示的代数式代入，直到 $i=m$，$E_{m,j}$ 只与 $E_{m,k}$ 有关，解出后回代即可求解。

由于每次只对 n 个方程进行消元，要处理的项不超过 $2n+1$ 个，这样时间复杂度可以达到 $O(n^3m)$，但是实现起来可以略显麻烦。所以我们就可以思考如何直接应用到高斯消元的过程之中。

同样为了避免回代，可以以逆序也就是 $E_{m,n}$，$E_{m,n-1}$,…，$E_{m,1}$，$E_{m-1,n}$，$E_{m-1,n-1}\cdots$，$E_{m-1,1}$,…，$E_{1,n}$，$E_{1,n-1}$,…，$E_{1,1}$ 这样一个顺序消元。

若当前消去的元为 $E_{x,y}$。而到 $E_{x,y}$ 时为第$((m-x+1)\times n-y+1)$步，按照高斯消元的过程，需要接下来的$(m\times n-((m-x+1)\times n-y+1))$也就是$((x-1)\times n+y-1)$个方程进行操作。或者说与开始的 mn 个方程相比，减少的方程数和消去的未知量数相等。对于所有满足 $i<x-1$ 或 $i=x-1$ 且 $j<y$ 的方程 $E_{i,j}=P_{i,j}^{(1)}E_{i+1,j}+P_{i,j}^{(2)}E_{i,j+1}+P_{i,j}^{(3)}E_{i-1,j}+P_{i,j}^{(4)}E_{i,j-1}+1$ 是不包含 $E_{x,y}$ 及之前消去的未知量的，所以它们并未改变，即不属于减少的方程，且与本次消元也无关。满足以上条件的方程即增广矩阵中不需要有变化的行有$((x-2)\times m+y-1)$个。所以相减后可以发现，每次消元时，实际需要进行消元的有关的方程不超过 n 个。而可以看出，由于 $E_{x,y}$ 之前项的系数已被消去，而对于这其中 n 个方程中位于消元顺序最后的且系数大于 0 的未知量为 $E_{x-2,y}$，所以这 n 个方程中系数大于 0 的项不超过 $2n$ 个。

这样一来，每次消元只需要与其后至多 n 行中的 $2n+1$ 列进行处理进行即可。时间复杂度为 $O(n^3m)$，若对于第 i 行的某个格子列出的方程只储存第 i-1 行至第

i+1 行的格子对应的未知量系数及常数，空间复杂度可以达到 $O(n^2m)$。这样本题得到一个不错的解决。

例题五：**Mario**[7]

Mario 生活在一个 $N \times M$ 的迷宫里，格子的属性分别为金币，怪兽，管道，墙和空地。每当 Mario 走到金币的格子上时，它都能拿到所给出的金币数（但金币不消失）。当 Mario 进入怪兽所在格子时，他将损失一条命。当 Mario 进入管道的入口时，他会立即传送到管道的出口，出口唯一，但是对应出口的入口可能有很多。墙是不可进入的。管道的出口和空地一样，进入时不会有任何事情发生。

Mario 从一个给定的起点（起点在空地上）开始，他有 3 条命，他每次等概率向相邻 4 格能走的格子走，在命为 0 或者他不能再获得更多金币时游戏结束。问他能得到的金币的期望值，若期望值为无穷大则输出-1。

$1 \le N, M \le 15$。

分析：

首先对于这种 3 条命的游戏一般做法即对图进行分层，对应了剩余命分别为 3，2，1 条。除了障碍和管道入口每个格子都对应了图中每层各一个点。对于 1 条命图中的怪兽对应的顶点为结束顶点，k ($k > 1$)条命对应的点分别对应 $k-1$ 条命对应层能到达的顶点连边。其他格子与其能到达的格子同层连边，对于有金币的格子对应的点权为金币数，其他格子对应的点权为 0。

但是这个游戏还有另外一种结束条件，即不能获得更多金币，那么对于所有格子进行广度优先搜索，若其不能到达一格金币数大于 0 的格子那么这个格子满足这个条件。即满足条件的格子都作为结束结点。搜索时不必判断命的情况，因为若不能在命数限制下到达那么在分层图中也是不连通的。对于这类格子在图中

[7] 题目来源：MIT 1st Team Contest 2007 (SPOJ 2324)

的结点也没有发出的边。不过实际编写程序时的时候可以不将这些顶点放入图内，在方程中按照 0 处理。

这样处理之后一样地列出方程求解。而当然，实际操作时不必构图，而可以直接在地图上建立方程组。

若不能求出起点的期望（即没有唯一解的情况下），那么此时由于图中不存在拿不到金币的点，且所有点权非负，所以可以判断出期望就为如题目描述中所说的无穷大。

还有需要注意的一点，这题就存在起点不能到达的点，可以利用判断某个格子是否能走到能拿到钱的格子的广度优先搜索过程处理出能到达的点，这样既不会增加编程复杂度又能节省常数时间。

总结

本文对竞赛中出现利用递推也包括动态规划解决的几道问题说起，对每题的特点总结出一类问题的状态设计方法和一般的解法。之后又对无法用递推有效解决的一个图模型，由于迭代的效率低，且无法得到精确解，提出了建立线性方程组并利用高斯消元解决的方法。而题目不会是一成不变的，所以随后通过两道例题讲解了实际应用到具体题目时的优化及处理。

当然，不能说本文覆盖了解决信息学竞赛所有期望值求解问题的有效方法。然而把握期望的概念，灵活设计方法，注意观察题目的特点，从而发现本质，这才是最为关键的。

参考文献：

[1] http://www.wikipedia.org/ Wikipedia

[2] 2004 年国家集训队研究报告 《有关概率和期望问题的研究》 鬲融