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

预备知识

- [什么是数学期望]

  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 $E(X)=\sum_i p_i x_i$

预备知识

- [全期望公式]

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

***E*(*Y* | *X* = 2)=3**

***P*(*X* = 2)=0.4**

***E*(*Y* | *X* = 1)=4**

***P*(*X* = 1)=0.6**

***E*(*Y*)=0.4×3 + 0.6×4 = 3.6**

引言

一、利用递推或动态规划解决

二、建立线性方程组解决

- 模型
- 例题： First Knight
- 例题： Mario

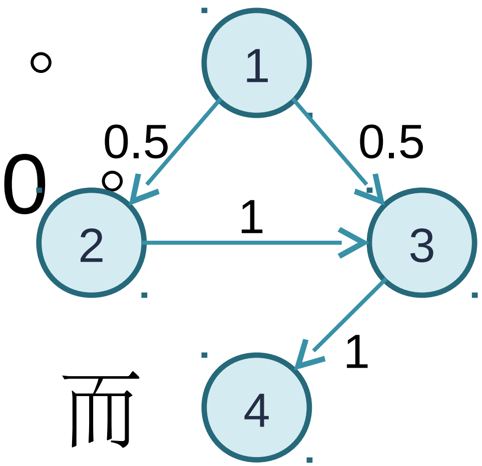
引入模型

- 给出一张有向图 $G = (V, E)$ 。
- 顶点 i 的权值为 W_i 。
- 给出 $P_{u,v}$ 表示顶点 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 ，那么在顶点 i 时有 $1 - P_i$ 的概率停止行动。
- 定义路径权为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，到某一顶点停止行动时所走的路径权的期望值。

引入模型

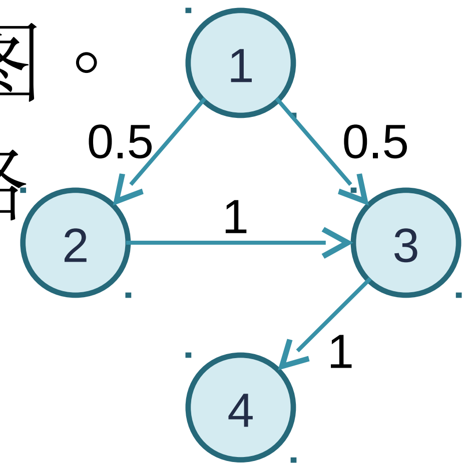
- 例如这张有向图，　$s = 1$ 。
- $W_1 = W_2 = W_3 = 1$，　$W_4 = 0$ 。
- 可以看到有两条路径。两条路径权分别为 3 和 2，而走这两条路径的概率均为 0.5 。
- 所以得到的期望为

$$2.5 = 0.5\times3 + 0.5\times2$$ 。



引入模型

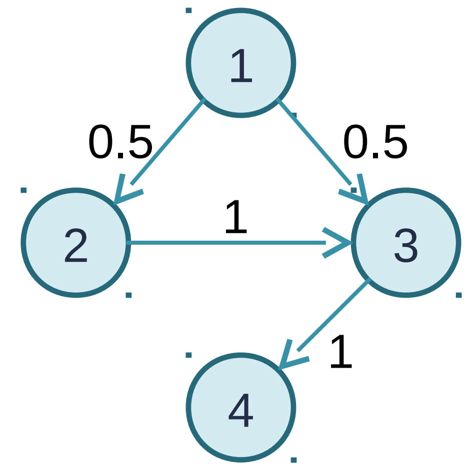
- 对于这种不存在环的有向图。
- 设 F_i 表示从顶点 i 出发的路径权期望。
- 可以分成两类情况。
- 从顶点 i 出发经过相邻顶点 k 的路径权期望为 F_k+W_i，概率 $P_{i,k}$。
- 停止行动路径权 W_i。



引入模型

- 可以得到如下的递推式
  并按照拓扑序来递推

$$F_i = \sum_{(i,k)\in E} P_{i,k} F_k + W_i$$

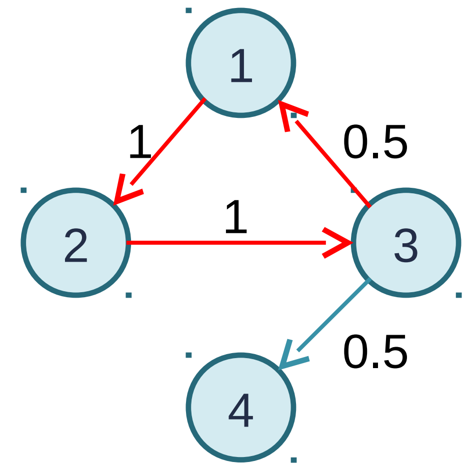


- 但若将这张有向图稍作修改

引入模型

- 可以得到如下的递推式并按照拓扑序来递推

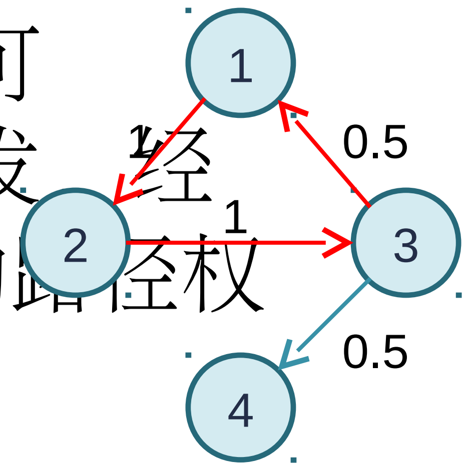
$$F_i = \sum_{(i,k)\in E} P_{i,k} F_k + W_i$$



- 但若将这张有向图稍作修改图存在环。

引入模型

- 所以对于一般的有向图，可以设 $F_{i,j}$ 为从顶点 i 出发，经过 j 步所走路径的路径权期望。



- 那么有：

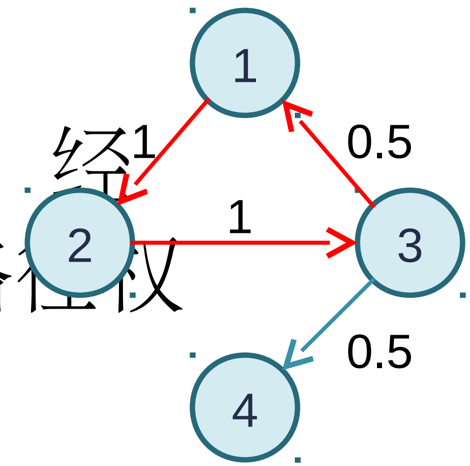
$$F_{i,0} = W_i$$

当 $j > 0$ 时

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

引入模型

- 所以对于一般的情况，可以设 $F_{i,j}$ 为从顶点 i 出发，经过 j 步所走路径的路径权期望。
- 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。



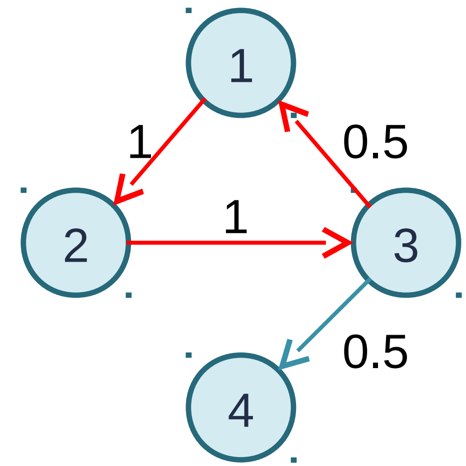
$$F_{i,0} = W_i$$

当 $j > 0$ 时

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

引入模型

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



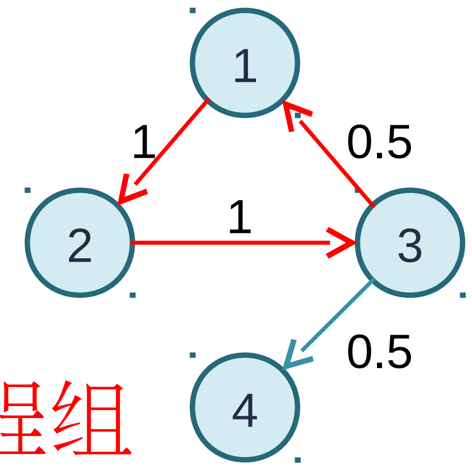
- 若 $F_{i,j}$ 当 $j\to\infty$ 时收敛，设收敛于 F_i

  那么答案即为 F_s 。
- 可以利用迭代求出满足精度要求的解，但是时间复杂度无法接受。

引入模型

- 方程形式：

$$F_i = \sum_{(i,k)\in E} P_{i,k} F_k + W_i$$

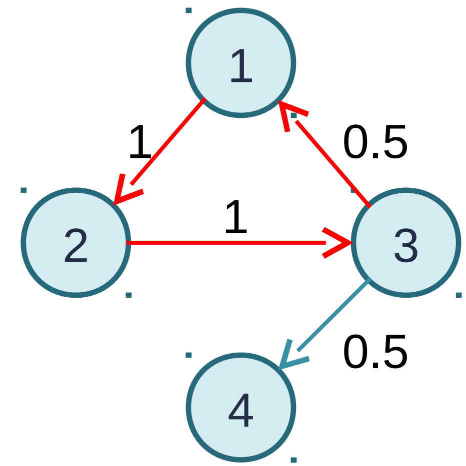


- 对于右图可以得到如下方程组

$$\begin{cases} F_1 = F_2 + 1 \\ F_2 = F_3 + 1 \\ F_3 = 0.5F_1 + 0.5F_4 + 1 \\ F_4 = 0 \end{cases}$$

引入模型

$$F_i = \sum_{(i,k)\in E} P_{i,k} F_k + W_i$$



- 高斯消元

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 1 \\ -0.5 & 0 & 1 & -0.5 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

引入模型

- 方程组中只含有与 s 相关的点。
- 方程组没有唯一解的情况。
- 可以调整消元顺序让所要求的 F_s 放在最后，这样就可以不用回代。
- 若权在边上而不在点上的话，设边 (u, v) 的权值为 $W_{u,v}$，那么同理方程即为

$$F_i = \sum_{(i,j)\in E} P_{i,j}(F_j + W_{i,j})$$

例题： First Knight

[问题描述]

- 题目来源： SWERC 08
- 一个 $m \times n$ 的棋盘，左上至右下编号为 (1, 1) 至 (m, n)， 并给定每个格子到周围四个格子的概率 $P_{i,j}^{(k)}$ 。
- 一个骑士从 (1, 1) 开始，按照给定概率走，问到达 (m, n) 的期望步数。
- 题目保证从任一格开始到 (m, n) 的概率均为 1 。

例题： First Knight

[分析]

- 列出方程直接求解?
- $E_{i,j}$ 表示从 (i, j) 出发的步数期望。

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

$m, n \le 40$ Accept?

时间复杂度 $O(m^3n^3)$

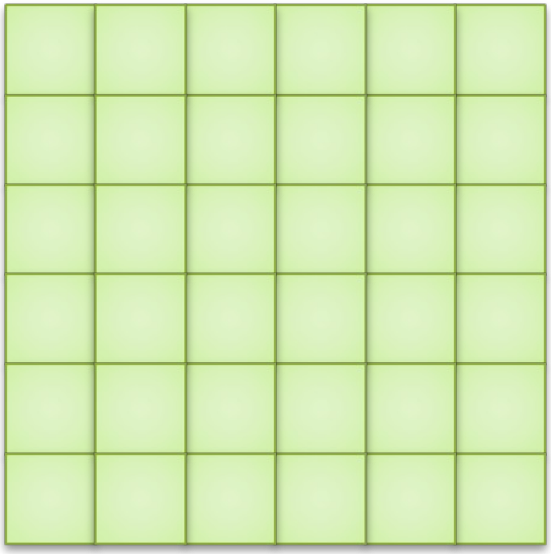
进行优化！

Time limit exceeded ☹

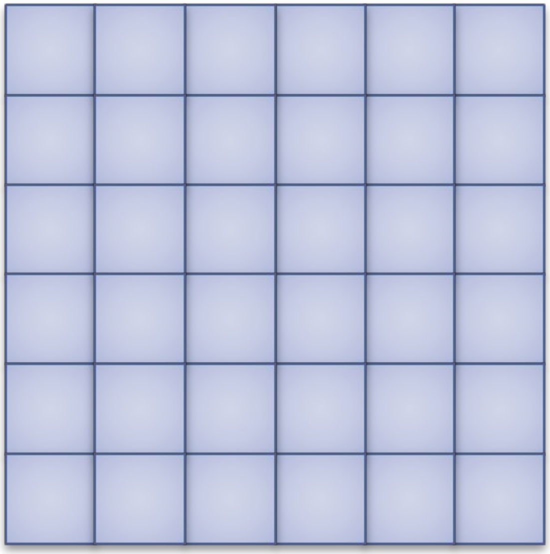
例题： First Knight [优化]

第 i 行第 j 列的格子表示了方程：

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



方程

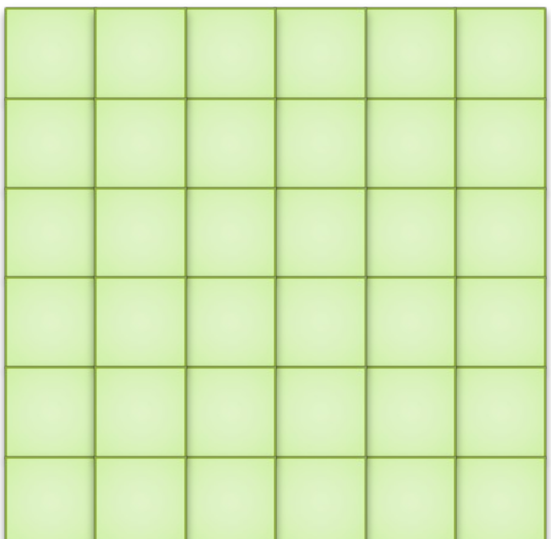


未知量

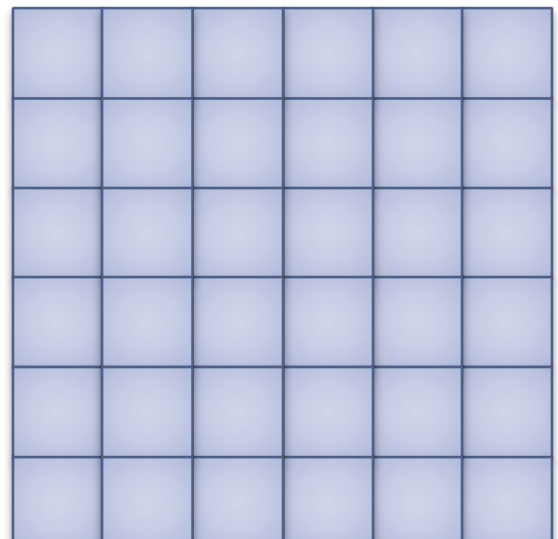
例题： First Knight [优化]

第 i 行第 j 列的格子表示了未知量：

$E_{i,j}$



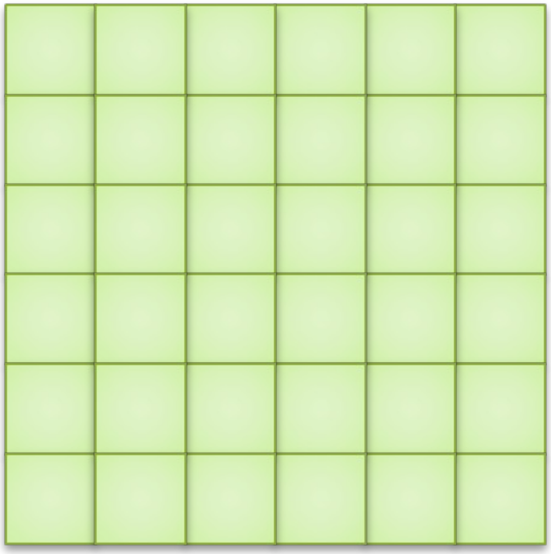
方程



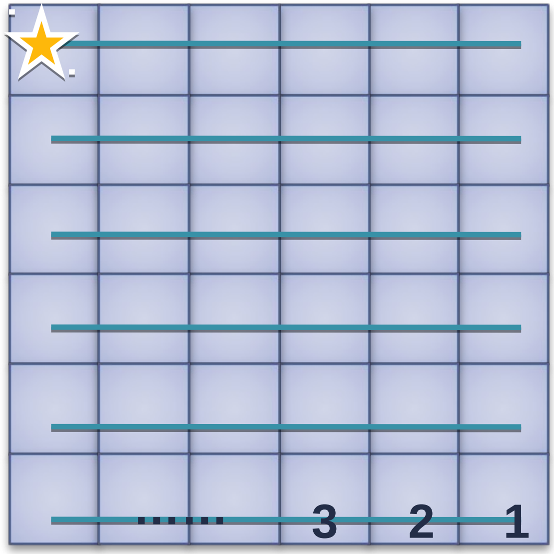
未知量

例题： First Knight [优化]

- 同样为了避免回代，可以逆序也就是 $E_{m,n}$ 到 $E_{1,1}$ 的顺序进行消元。



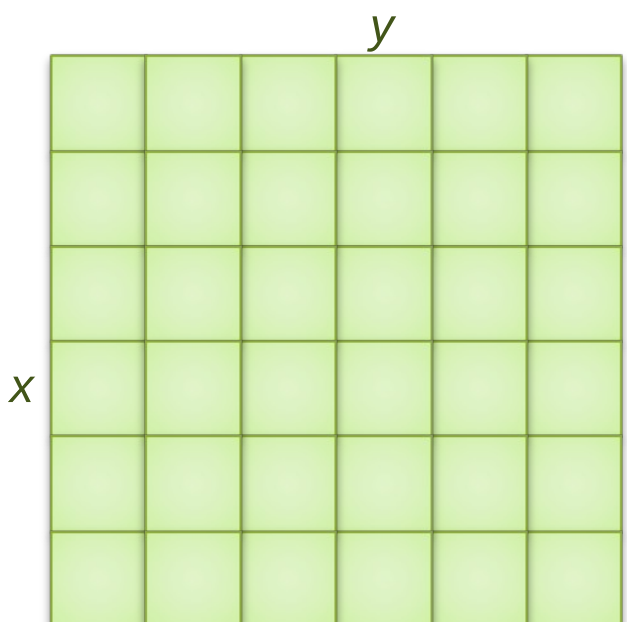
方程



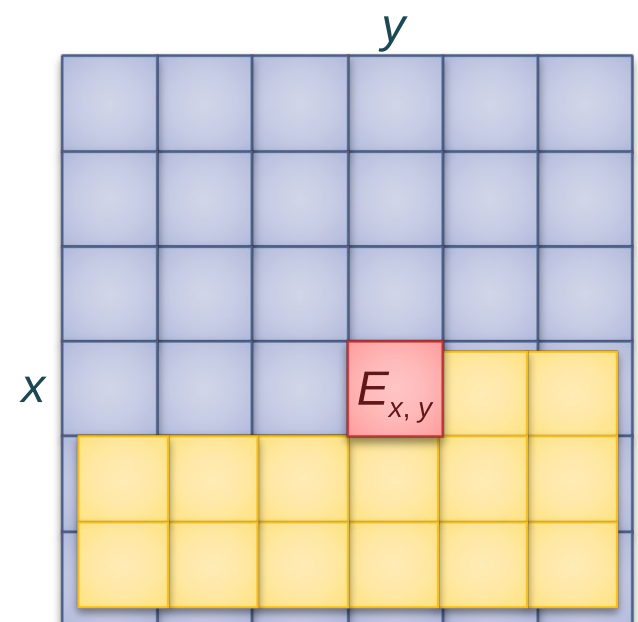
未知量

例题： First Knight [优化]

- 对于方程而言，若当前要消去的未知量为 $E_{x,y}$ 。



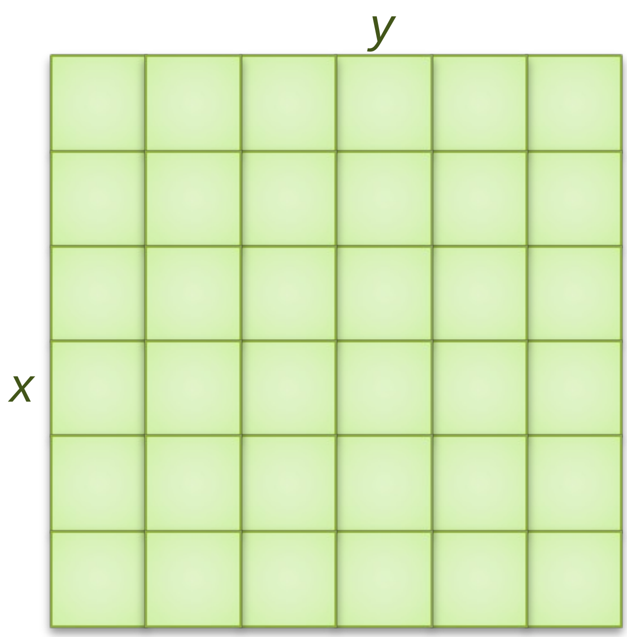
方程



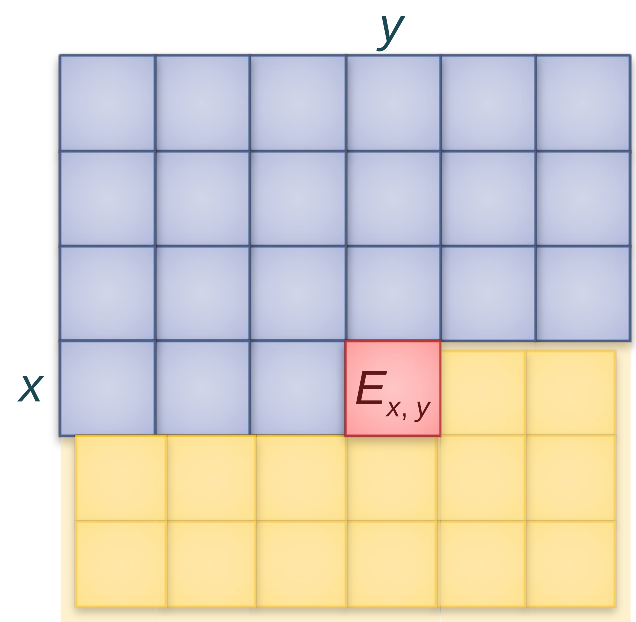
未知量

例题： First Knight [优化]

- 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。



方程

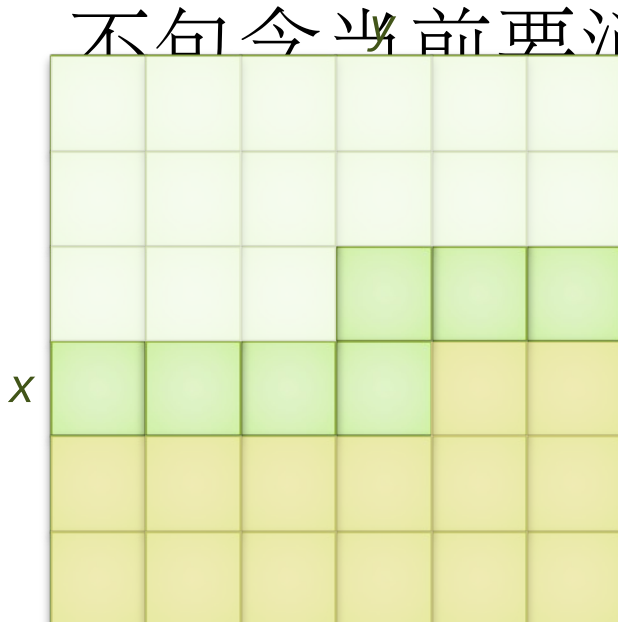


未知量

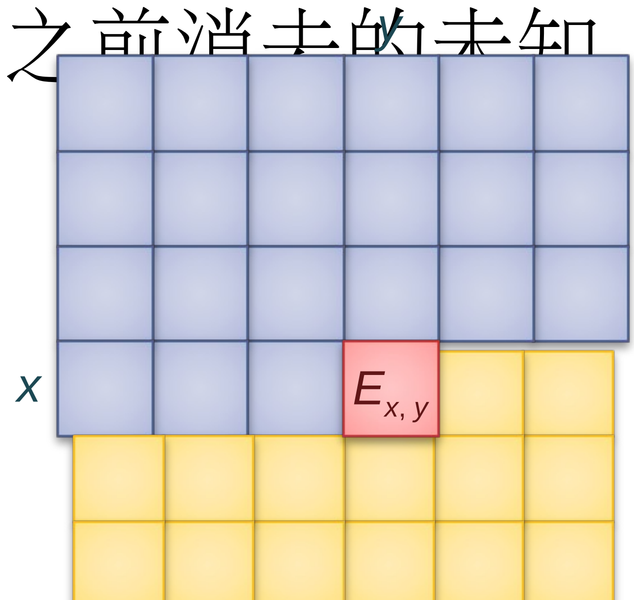
例题： **First Knight** [优化]

- 满足 $i < x - 1$ 或 $i = x - 1$ 且 $j < y$ 的方程 $E_{i,j} \quad P^{(1)}_{i,j}E_{i+1,j} + P^{(2)}_{i,j}E_{i,j+1} + P^{(3)}_{i,j}E_{i-1,j} + P^{(4)}_{i,j}E_{i,j-1} + 1$

不包含当前要消的和之前消去的未知



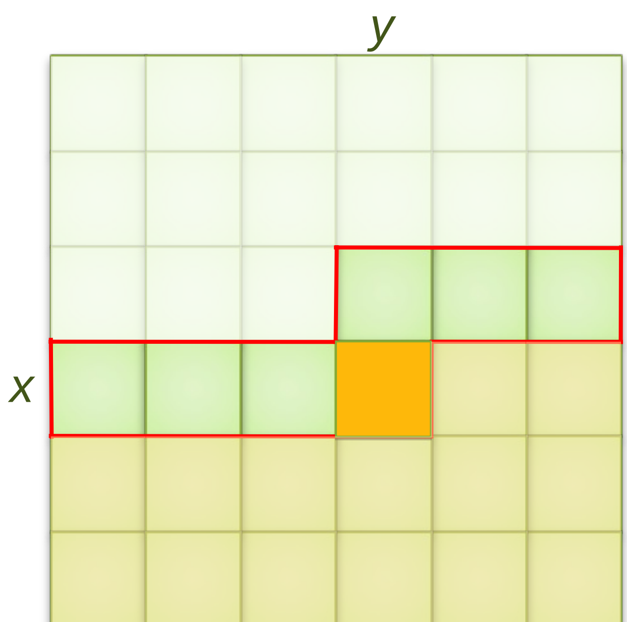
方程



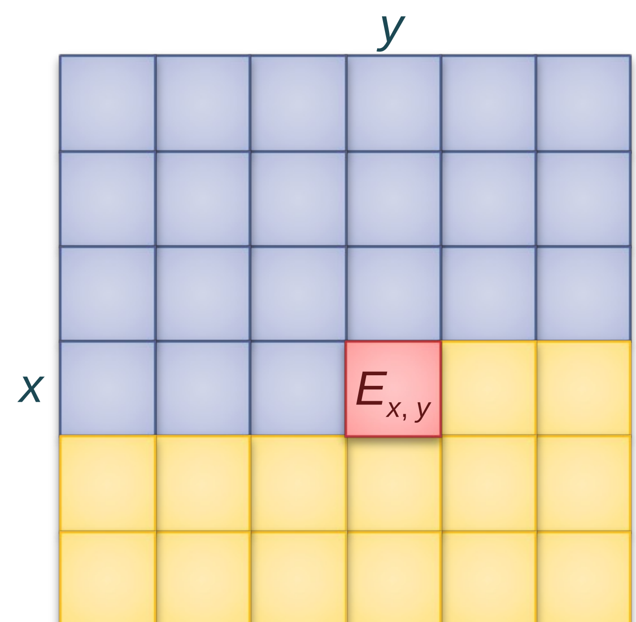
未知量

例题： First Knight [优化]

- 所以最多与 n 个方程进行消元。



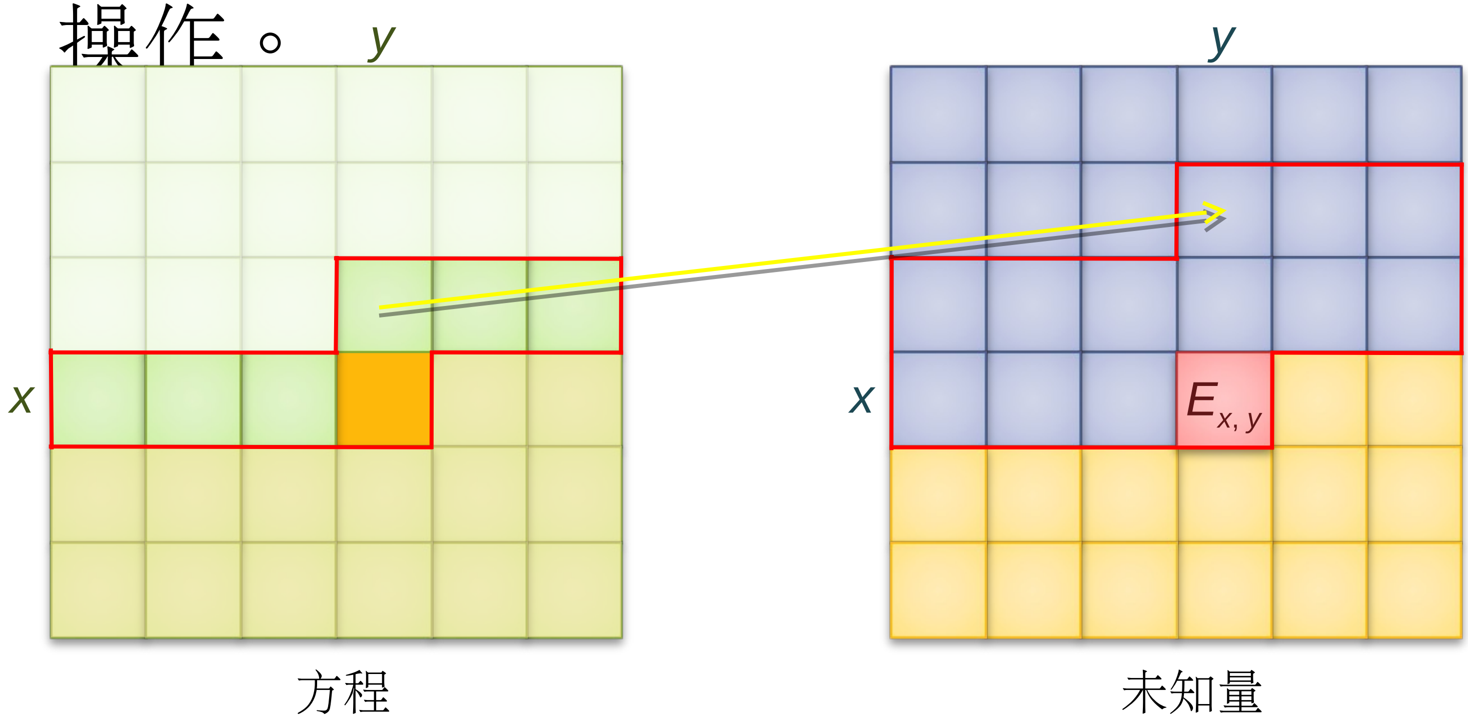
方程



未知量

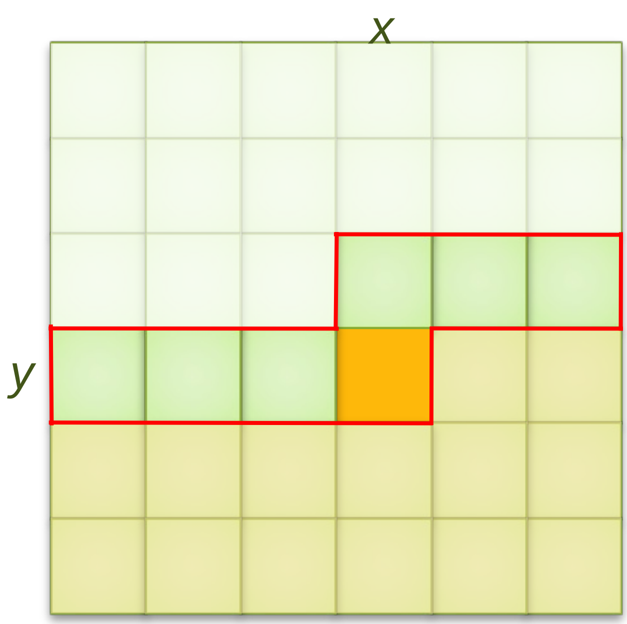
例题： First Knight [优化]

- 消元顺序最后的未知量为 $E_{x-2,\,y}$。
- 所以对于增广矩阵来说，每次消元最多只需要对 n 行和其中的 $2n+1$ 列进行操作。

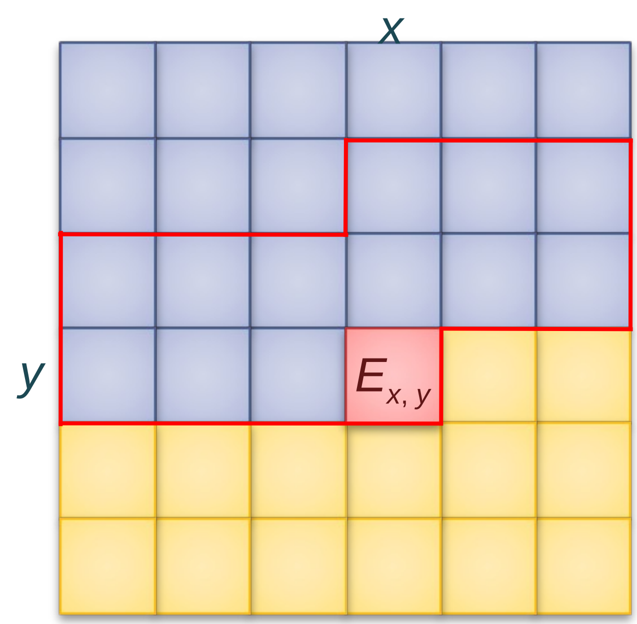


例题： First Knight [时空复杂度]

- 时间复杂度 $O(n^3m^3) \rightarrow O(n^3m)$ 。
- 空间复杂度可优化至 $O(n^2m)$ 。

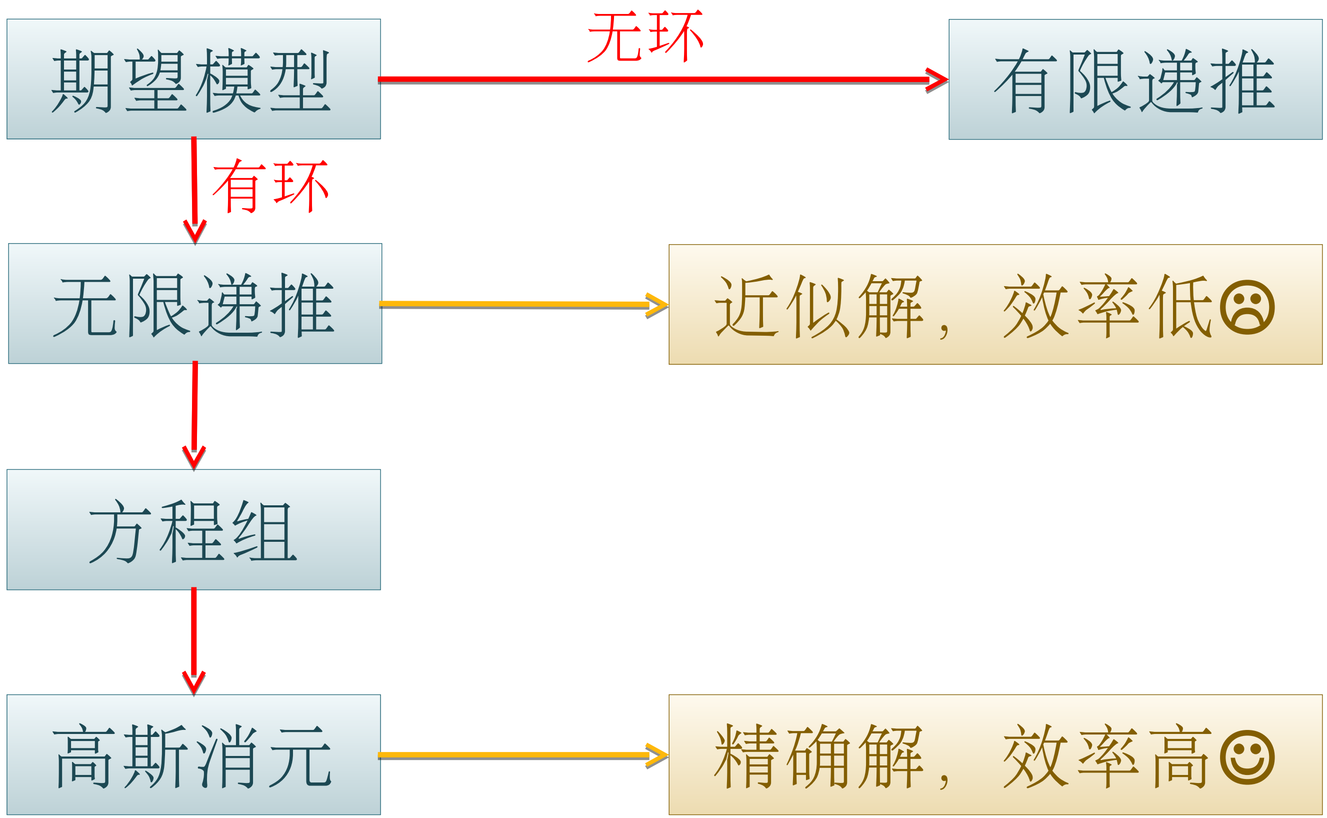


方程



未知量

总结



总结

典型模题

对应

总结

具体问题

$\downarrow$ 对应

期望模型

$\downarrow$ 选择

解决方法

$\downarrow$ 特点

处理优化

谢谢