

《网格图最大流计数》解题报告

北京市十一学校 吴畅

1 题目描述

给定 n, m, k ，和两个正整数序列 $a_{1\dots n}, b_{1\dots m}$ ，以及一个 $k \times k$ 的 01 矩阵 $s_{1\dots k, 1\dots k}$ 。

考虑一张有向图 $G = (V, E)$ ，其中 $V = \{S, T\} \cup (\{0, 1\} \times ([1, k] \cap \mathbb{Z})^2)$ ，而 $E = E_1 \cup E_2 \cup E_3$ 由三部分组成：

- $E_1 = \{(S, (0, 1, a_i)) \mid 1 \leq i \leq n\} \cup \{((1, k, b_i), T) \mid 1 \leq i \leq m\}$
- $E_2 = \{((1, i, j), (0, i + 1, j)) \mid 1 \leq i < k, 1 \leq j \leq k\} \cup \{(1, i, j), (0, i, j + 1)) \mid 1 \leq i \leq k, 1 \leq j < k\}$
- $E_3 = \{((0, i, j), (1, i, j)) \mid 1 \leq i, j \leq k, s_{i,j} = 1\}$

简单来说，你可以看成每个格子 $(i, j), 1 \leq i, j \leq k$ 被拆成了一个入点 $(0, i, j)$ 和一个出点 $(1, i, j)$ 。 E_1 描述了 S, T 与这些点之间的边，由 a, b 决定； E_2 描述了每个格子的出点连向它上方和右方格子的入点的边； E_3 描述了每个格子的入点连向出点的边，由 s 决定。

现在我们将 G 看成一个网络，每条边的容量是 1。你要求出以 S 为源点，以 T 为汇点的最大流，以及最大流的数量（两个流被认为是不同的，当且仅当存在一条边在两个流中的流量不同）。

2 数据范围与约定

对于全部测试点， $n, m \leq k \leq 400$ 。

时间限制：2.5 s

空间限制：512 MB

具体子任务限制如下表所示。

子任务编号	$n \leq$	$k \leq$	特殊性质	子任务分值
1	7	7	无	5
2	18	18		5
3	10	400		10
4	100			25
5	400		$n = m$ 且最大流为 n	10
6			最大流为 n	25
7		无	20	

3 解题过程

3.1 算法一

我会暴力！

直接爆搜可以通过第一个子任务。

注意到流可以看成若干条不交路径。考虑逐列轮廓线 dp。

时间复杂度 $O(k^2 2^k)$ ，可以通过第二个子任务。

3.2 算法二

我会 LGV 引理！

当 $n = m$ 且最大流为 n 时，相当于给定若干起点和终点求不相交路径。先算出每对起点和终点间的路径方案数，这只需要枚举起点然后做一遍 $O(k^2)$ 的 dp，最后用 LGV 引理转化成求行列式即可。

时间复杂度 $O(nk^2 + n^3)$ 。可以通过第五个子任务。

3.3 算法三

我会手动 LGV！

当 m 变大时，算法二的复杂度瓶颈在于枚举终点集合。

考虑直接在终点上按顺序 dp。设 $dp_{i,S}$ 表示考虑了前 i 个终点，它们已经匹配了起点集合 S 的所有方案的贡献。

转移可以钦定一个还未匹配过的起点和第 $i + 1$ 个终点匹配，乘上路径方案数以及对逆序对奇偶性的影响。

这个做法本质上是对 LGV 展开后的每一种匹配方式 dp。

时间复杂度 $O(2^n nk)$ 。可以通过第三个子任务。

3.4 算法四

我会 Pfaffian!

考虑最大流为 n 的情况，问题相当于是固定了起点但不固定终点的不相交路径。

注意到 LGV 引理中实现双射抵消的工具是行列式自带的排列奇偶性，我们想办法动用人类智慧将这个东​​西转化成只和 n 个起点有关。

先来介绍一个代数工具：Pfaffian。

定义斜对称矩阵 A 为满足 $A = -A^T$ 的方阵，注意其主对角线上的元素必须为 0。

定义 $\mathfrak{S}_{2n}$ 为所有满足下述条件的 $1, 2, \dots, 2n$ 的排列 P 的集合：

- $\forall 1 \leq i \leq n, P_{2i-1} < P_{2i}$
- $\forall 1 \leq i < n, P_{2i-1} < P_{2i+1}$

可以发现，如果将 P_{2i-1} 和 P_{2i} 看成一对匹配的元素，实际上 $\mathfrak{S}_{2n}$ 包含的就是所有本质不同的完美匹配。（下文中匹配指一对元素，完美匹配指所有 n 对匹配）

定义斜对称矩阵 $A = (a_{i,j})_{1 \leq i,j \leq 2n}$ 的 **Pfaffian** 为 $\text{Pf}(A) := \sum_{\pi \in \mathfrak{S}_{2n}} \text{sgn } \pi \prod_{1 \leq i \leq n} a_{P_{2i-1}, P_{2i}}$ 。

其中 $\text{sgn } \pi = (-1)^{\sum_{1 \leq i < j \leq 2n} [\pi_i > \pi_j]}$ 。

Pf 和 det 有非常紧密的联系。这里给出两个常见结论（ B 为任意方阵）：

- $\text{Pf}(A)^2 = \det(A)$
- $\text{Pf}(BAB^T) = \text{Pf}(A)\det(B)$

对于给定的斜对称矩阵 A ，为了计算 $\text{Pf}(A)$ ，你可能会考虑用第一条结论转化成求 $\det(A)$ 。可惜在常见的模数域上开根号并不是一件方便的事情。但事实上，我们只需要令第二条结论中的 B 是初等行变换矩阵，就可以得到与 det 类似的消元过程，更详细的内容此处不再赘述。复杂度同高斯消元，为 $O(n^3)$ 。

回到原问题上，我们初步的想法是枚举 n 个起点的一个完美匹配，通过上面对完美匹配的奇偶性的定义，复刻“交换最小的交点前的路径”这一操作。

这个想法刚好和 Pfaffian 的定义吻合，因此原问题的答案为 $\text{Pf}_{1 \leq i < j \leq n}(Q(i,j))$ （这个记号表示上三角部分用 $Q(i,j)$ 描述的斜对称矩阵的 Pfaffian），其中 $Q(i,j)$ 表示所有第 i 个起点和第 j 个起点组成的不交路径对数量。

事实上这里还有若干细节需要验证，比如一个合法的路径的贡献恰好为 1，双射抵消的两个排列奇偶性相反等等，大多比较简单，限于篇幅此处略去。

最后还需要注意一个问题，上面的方法只解决了 n 是偶数的情况。如果 n 是奇数，取一个不能到达任何点的虚点，同时加入起点集合和终点集合，便可以转化为 n 是偶数的情况。

时间复杂度 $O(nk^2 + n^3)$ 。可以通过第六个子任务。

3.5 算法五

我会生成函数和插值！

当起点也不固定时，我们可以考虑取一个形式元 x ，每选择一个起点就乘上一个 x ，这样最终就可以得到不相交路径数量关于选择的起点个数的生成函数。

这个过程可以看成，还是先枚举一个完美匹配，但是现在每个匹配中的两个起点都可以同时被选上或者同时不被选上，如果同时被选上就要贡献 x^2 。

但是需要解决没被选的匹配对逆序对奇偶性的影响。我们已经知道一个集合内部任意匹配的总贡献是 1。假设被选上的匹配集合是 S ，没被选上的匹配集合是 T （满足 $S \cup T = \{1, 2, \dots, 2n\}, S \cap T = \emptyset$ ）。现在的问题是，我们想将 S 和 T 之间的逆序对独立出来，以便我们给单个匹配分配合适的系数。

注意到，从任何一个完美匹配 π 中移除一对匹配 (i, j) ，对逆序对奇偶性的影响恰好是 $i + j + 1$ ，因为 i, j 之间的点要么和外面的点匹配，贡献一次；要么互相匹配，贡献零次。进一步，移除一个匹配集合 T 的影响是 $\frac{|T|}{2} + \sum_{i \in T} i$ 。这样一来，我们便可以保证每个合法方案的贡献恰好为 1。

另一个问题是若干个匹配包含的起点数量一定是偶数。如果我们想选奇数个起点，需要添加一个虚点和其中一个起点匹配，匹配被选上的系数就是从这个起点出发的路径数量。因此原问题的答案

当 n 是奇数时

$$= \text{Pf}_{1 \leq i < j \leq n+1}((-1)^{i+j+1} + x^2 Q(i, j))$$

其中 $Q(i, j), j \leq n$ 的取值和算法四相同， $Q(i, n+1) = x^{-1}V(i)$ ， $V(i)$ 表示第 i 个起点的路径数量。

当 n 是偶数时，为了让斜对称矩阵的大小变成偶数，我们人为添加一个空点，它和任何其它起点被同时选上的系数都是 0

$$= \text{Pf}_{1 \leq i < j \leq n+2}((-1)^{i+j+1} + x^2 Q(i, j))$$

其中 $Q(i, j), j \leq n+1$ 的取值和上式相同， $Q(i, n+2) = 0$ 。

剩下的问题便是将 Pfaffian 的值算出来。直接做会涉及多项式的四则运算，并不好处理。一个经典的方法是插值，用正常的消元过程算出答案的 n 次多项式在 $n+1$ 个点处的点值，最后再把系数求出来。

时间复杂度 $O(nk^2 + n^4)$ 。可以通过第四个子任务。

3.6 算法六

我会求 $\text{Pf}(\mathbf{A} + \mathbf{B}x + \mathbf{C}x^2)!$

上述问题实质上就是求每个元素为二次多项式的 Pf 值。注意到如果我们先求出 $\det$ ，是有办法甄别两个根中真正的那个的，因为其常数项一定为 1。

因此不妨转化为求 $\det(\mathbf{A} + \mathbf{B}x + \mathbf{C}x^2)$ ，这是一个经典问题，我们在此简述一下它的经典做法。

首先我们可以不断消元使 $\mathbf{C}$ 变为单位矩阵，注意当 $\mathbf{C}$ 不满秩时（某一行全为 0）行列式仍可能不为 0。只需要将这一行同时乘上 x ，最后再除掉 x 。如果 $\mathbf{A}, \mathbf{B}, \mathbf{C}$ 在这一行都为 0，那么行列式才为 0。

现在问题形如求 $\det(\mathbf{I}x^2 - \mathbf{F}_1x - \mathbf{F}_2)$ 。我们知道给定矩阵 A 求其特征多项式 $|\lambda\mathbf{I} - \mathbf{A}|$ 是容易的。由行列式的一些基本结论，容易发现我们有

$$\det(\mathbf{I}x^2 - \mathbf{F}_1x - \mathbf{F}_2) = \det\left(x\mathbf{I} - \begin{bmatrix} & \mathbf{I} \\ \mathbf{F}_2 & \mathbf{F}_1 \end{bmatrix}\right)$$

因此问题转化成求矩阵的特征多项式。可以利用相似矩阵的特征多项式相等的性质，消成上海森堡矩阵，然后进行一次 dp 求出答案。

时间复杂度 $O(nk^2 + n^3)$ 。可以通过全部子任务。