浅谈矩阵在信息学竞赛中的应用

四川省成都市第七中学 杨宁远

摘 要

矩阵在信息学中有着极为广泛的应用。本文从矩阵的基本概念和性质出发，针对 LGV 引理、特征多项式等问题给出了优秀的算法，并讨论了若干道与之相关的信息学竞赛例题。本文还介绍了余子式及其高效的计算方法，探讨了余子式在矩阵树定理中的应用。在最后，本文介绍了计算特征值数值解有关的方法，例如 Jacobi 算法和 QR 迭代法等。

1 引言

矩阵是线性代数的重要部分，与之相关的线性代数工具有行列式、余子式、特征多项式、特征值与特征向量等等。这些工具可以有效帮助我们解决信息学竞赛中的很多问题，为我们提供更为本质的视角。

笔者在本文中梳理和总结了矩阵在信息学中的应用，希望能借此机会，向选手们普及矩阵相关的知识和算法，吸引选手们进行更进一步的研究。笔者也希望在将来的算法领域里，能看到更多与线性代数有关的题目及其优美的解法。

本文的第 2 节为矩阵相关的记号约定；第 3 节介绍了行列式以及由此推导出来的 LGV 引理；第 4 节介绍了余子式、伴随矩阵及其求法和矩阵树定理；第 5 节介绍了矩阵的线性递推及其应用；第 6 节介绍了矩阵的特征值、特征多项式和相关的应用；第 7 节讨论了特征值的一些求解方式。

2 记号约定

2.1 矩阵的定义

一个由 $n \times m$ 个数排列为 n 行 m 列的数表被称为一个 n 行 m 列的矩阵，简称为一个 $n \times m$ 的矩阵。

对于一个 $n \times m$ 的矩阵 A，它可以写作：

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,m} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,m} \end{bmatrix}$$

其中矩阵第 i 行 j 列的元素被记为 $a_{i,j}$。

特殊地，若 $n = m$，那么我们可以称 A 为 n 阶矩阵或方阵。

2.2 矩阵的基本运算

定义 2.1 (加法). 两个 $n \times m$ 的矩阵 A 与 B 的加法定义为其元素对位相加。令 $C = A + B$，那么有 $c_{i,j} = a_{i,j} + b_{i,j}$。

定义 2.2 (减法). 两个 $n \times m$ 的矩阵 A 与 B 的减法定义为其元素对位相减。令 $C = A - B$，那么有 $c_{i,j} = a_{i,j} - b_{i,j}$。

定义 2.3 (乘法). 一个 $n \times s$ 的矩阵 A 乘上另一个 $s \times m$ 的矩阵 B 会得到一个大小为 $n \times m$ 的矩阵 C，并且有：

$$c_{i,j} = \sum_{k=1}^{s} a_{i,k} \times b_{k,j}$$

定义 2.4 (数乘). 一个 $n \times m$ 的矩阵 A 乘上另一个数 x 的结果 $B = xA$ 有:

$$b_{i,j} = a_{i,j}x$$

定义 2.5 (转置). 对于一个 n 阶矩阵 A，其转置 A^T 被定义为将 A 中的元素按对角线翻转。也就是说如果 $B = A^T$，那么有 $b_{i,j} = a_{j,i}$。

3 行列式

3.1 行列式的定义与计算

定义 3.1 (行列式)**.** 对于一个 n 阶矩阵 A，其行列式被定义为：

$$|A| = \sum_{\sigma \in S_n} sgn(\sigma) \prod_{i=1}^{n} a_{i,\sigma_i}$$

其中 S_n 为所有 n 阶排列，$sgn(\sigma)$ 为 $(-1)^{\sigma\text{的逆序对个数}}$。

显然，如果我们按照定义直接枚举排列计算行列式的复杂度是 $O(n!)$ 的。然而这种方法效率过于低下，大部分时候都不能满足我们的要求。

定义 3.2 (矩阵的初等变换)**.** 矩阵的初等变换定义为：

1. 将第 i 行加上第 j 行的 k 倍（$i \neq j$）

2. 将第 i 列加上第 j 行的 k 倍（$i \neq j$）

3. 将第 i 行乘上 k（$k \neq 0$）

4. 交换 i, j 两行。

定理 3.1. 对矩阵进行第 $1, 2$ 种初等变换不改变行列式的值，第 3 种会将行列式乘上 k，第 4 种会将行列式乘上 -1。

证明. 将行列式展开后容易发现第 $1, 2$ 种的贡献会被抵消掉，而第 3 种会使得所有排列的答案全部乘上 k。

对于第 4 种，我们发现如果交换排列中的两个数，那么逆序对数奇偶性会变化。 □

综上，我们可以通过矩阵的初等变换，利用解方程时常用的高斯消元法来求解行列式。这种方法的时间复杂度为 $O(n^3)$。

也可以把高斯消元的过程看作左乘行变换矩阵或者右乘列变换矩阵来得到上三角矩阵。即 $P_1P_2P_3 \cdots P_kA = A' = AQ_1Q_2Q_3 \cdots Q_k$。

3.2 行列式的一些性质

定理 3.2.

$$|A| = |A^T|$$

证明. 由于排列求逆后不改变逆序对个数，而转置后的行列式可以看作把原矩阵的排列求逆，于是转置后的行列式不变。 □

定理 3.3.

$$|AB| = |A||B|$$

证明. 令 A', B' 分别为 A, B 进行高斯消元后的上三角矩阵。

将 A, B 分别表示为 $P_1P_2P_3\cdots P_kA'$，$B'Q_1Q_2Q_3\cdots Q_k$，那么也就有：

$$|AB| = |P_1P_2P_3\cdots P_kA'B'Q_1Q_2Q_3\cdots Q_k| = |A'B'| = |A'||B'|$$

□

3.3 LGV 引理

定理 3.4 (LGV 引理)**.** 对于一张有向无环图 G，边有边权。考虑一个大小为 n 的起点集合和一个大小为 n 的终点集合，定义 $e(a, b)$ 为所有从 a 到 b 的路径的边权乘积之和。定义矩阵

$$M = \begin{bmatrix} e(a_1, b_1) & e(a_1, b_2) & e(a_1, b_3) & \cdots & e(a_1, b_n) \\ e(a_2, b_1) & e(a_2, b_2) & e(a_2, b_3) & \cdots & e(a_2, b_n) \\ e(a_3, b_1) & e(a_3, b_2) & e(a_3, b_3) & \cdots & e(a_3, b_n) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ e(a_n, b_1) & e(a_n, b_2) & e(a_n, b_3) & \cdots & e(a_n, b_n) \end{bmatrix}$$

定义一组不相交路径为 n 条路径 $P_1, P_2, \cdots P_n$，使得存在一个排列 σ，其中第 i 条路径为从 a_i 出发，到 b_{σ_i} 结束，且路径两两不存在公共点。这组路径的带符号权值被定义为

$$f(P_1, P_2, \cdots P_n) = sgn(\sigma)\prod_i e(i, \sigma_i)$$

那么有：

$$|M| = \sum_{P_1, P_2, \cdots, P_n} f(P_1, P_2, \cdots, P_n)$$

证明. 假如说一组路径有某两条存在交点，那么我们找到最小的 i，并在 i 最小的基础上找到最小的 j 使得 P_i, P_j 有交。那么我们找到它们最靠前相交的点，并将之后的路径交换一下，容易发现这会使得 $sgn(\sigma)$ 改变，而路径组的权值不变。将得到的新路径组与原来的路径组进行匹配。容易发现所有有交的路径组都能一一匹配，又由于路径的权值前面有一个 $sgn(\sigma)$，于是有交路径组的权值恰好能够两两抵消。

而对于不相交的路径，行列式中 $sgn(\sigma)$ 正好与路径权值中的相符。于是得证。 □

3.4 例题

给定 n, m, k, r, c, V，求出满足如下条件的 $n \times m$ 矩阵的数量模 998244353 的值：

1. 对所有 $1 \le i \le n, 1 \le j \le m$，有 $1 \le a_{i,j} \le k$
2. 对所有 $1 \le i \le n, 1 \le j \le m-1$，有 $a_{i,j} \le a_{i,j+1}$
3. 对所有 $1 \le i \le n-1, 1 \le j \le m$，有 $a_{i,j} \le a_{i+1,j}$
4. $a_{r,c} = V$

$n, m \le 200, k \le 100$。

3.5 问题分析

题目来源：XXI Open Cup, Grand Prix of Tokyo, Problem A Ascending Matrix。

首先发现题目相当于要把 A 分成恰好 k 层，层可以为空。并且还规定了 $a_{r,c}$ 属于哪一层。要求层的轮廓路径只能向右或向上走，且层的轮廓路径不能交叉。

再转化一下，变成：有 $k-1$ 条从 $(n, 0)$ 走到 $(0, m)$，要求只能向右和向上走的路径，路径之间有顺序，后面的路径不能走到前面的路径左边。并且 $a_{r,c}$ 左边要有 $v-1$ 条路径。

我们将第 i 条路径向右平移 $i-1$ 格，再向下平移 $i-1$ 格，现在限制就变成了路径不交了。而第 i 条路径的起点变成了 $n+i-1, i-1$，终点变成了 $i-1, m+i-1$。容易发现如果 σ 的逆序对数 > 0，那么路径一定有交。因此如果不考虑 $a_{r,c}$，我们就可以直接使用 LGV 引理求出方案数。

现在我们需要保证 $a_{r,c}$ 左边恰有 $v-1$ 条路径。我们将一条路径的权值改写为：

$$\begin{cases} 1, & \text{路径不经过 } a_{r,c} \text{ 的左边} \\ x, & \text{路径经过 } a_{r,c} \text{ 的左边} \end{cases}$$

那么容易发现答案恰为 $[x^{v-1}]|A|$。而显然 $|A|$ 最高为一个 n 次多项式，因此带入 n 个点值后使用拉格朗日插值即可在 $O(n^3 k)$ 复杂度内解决。

4 余子式与伴随矩阵

4.1 余子式的定义和性质

定义 4.1 (余子式)**.** n 阶矩阵 A 中一个元素 $a_{i,j}$ 的余子式 $M_{i,j}$ 定义为 A 删掉第 i 行和第 j 列后的矩阵的行列式。

定义 4.2 (代数余子式)**.** n 阶矩阵 A 中一个元素 $a_{i,j}$ 的代数余子式 $A_{i,j}$ 定义为 $(-1)^{i+j}M_{i,j}$。

定理 4.1 (拉普拉斯按行展开)**.**

$$\forall i, |A| = \sum_{j} a_{i,j}A_{i,j}$$

证明. 我们枚举 $\sigma_i = j$，那么不难发现

$$\sum_{\sigma_i=j} sgn(\sigma)\prod_{i=1}^{n} a_{i,\sigma_i} = a_{i,j}A_{i,j}$$

□

定理 4.2 (拉普拉斯按列展开)**.**

$$\forall i, |A| = \sum_{j} a_{i,j}A_{i,j}$$

证明. 由于实际上对 $|A|$ 按列展开就是对 $|A^T|$ 按行展开，而又有 $|A| = |A^T|$，于是证毕。 □

定义 4.3 (伴随矩阵)**.** 矩阵 A 的伴随矩阵定义为：

$$A^* = \begin{bmatrix} A_{1,1} & A_{2,1} & A_{3,1} & \cdots & A_{n,1} \\ A_{1,2} & A_{2,2} & A_{3,2} & \cdots & A_{n,2} \\ A_{1,3} & A_{2,3} & A_{3,3} & \cdots & A_{n,3} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ A_{1,n} & A_{2,n} & A_{3,n} & \cdots & A_{n,n} \end{bmatrix}$$

定理 4.3. 当 $|A| \neq 0$ 时，有 $AA^* = |A|I$

证明. 将式子展开后会发现其实就是拉普拉斯展开。 □

利用上面的性质我们可以在 $O(n^3)$ 的时间复杂度内求出伴随矩阵。

如果 $rank(A) = n$，那么会有 $A^* = |A|A^{-1}$。直接使用矩阵求逆就可以求出伴随矩阵。

如果 $rank(A) \leq n-2$，那么删掉一行一列后一定仍然线性相关，也就是说余子式一定为 0。

否则如果 $rank(A) = n-1$，我们找到一组不全为 0 的列向量 $\vec{p}$ 和行向量 $\vec{q}$，使得 $A\vec{p} = 0, \vec{q}A = 0$。那么对于 $q_r \neq 0, p_c \neq 0$，有 $A_{i,j} = \frac{q_i p_j}{q_r p_c}A_{r,c}$。

证明. 观察一下 (r, i) 的余子式以及 (r, c) 的余子式，我们发现可以利用初等变换将 $A_{r,c}$ 变为 $A_{r,i}$。不妨令 $i < c$，令 (r, c) 的剩余矩阵为 B，(r, i) 的为 C。那么我们将 B 的第 $i+1 \sim c-1$ 列向左平移一格，将第 i 列交换到第 $c-1$ 列上。由于有 $\sum_i p_i A_i = 0$，其中 A_i 是 A 的第 i 列，于是我们将 B 的第 $i-1$ 列乘上 p_i，将剩余的列按 p 的系数加到第 $i-1$ 列上，最后除以 $-p_c$ 就可以得到 C 的第 $i-1$ 列。于是我们就有 $A_{r,i} = \frac{p_i}{p_c} A_{r,c}$。

对行来说也有类似的结论。将两者结合起来即得证。 □

4.2 矩阵树定理

定理 4.4 (矩阵树定理)**.** 对于一个无向图 G，它的生成树个数为：度数矩阵减去邻接矩阵删去第 i 行 i 列的行列式。

定理 4.5 (矩阵树定理的有向形式)**.** 对于一个有向图 G，它的以 i 为根的外向生成树个数为：入度矩阵减去边矩阵删去第 i 行 i 列的行列式。

类似地，它的以 i 为根的内向生成树个数为：出度矩阵减去边矩阵删去第 i 行 i 列的行列式。

4.3 例题

给定一张大小为 n 的有向图。对于每个点 i 求出以 i 为根的外向树的个数。

$n \leq 500$。

4.4 问题分析

题目来源：HDU 多校第 10 场 J 题。

这道题就是要求出所有的 $A_{i,i}$，直接用我们上面所讲的方法求出伴随矩阵即可。

复杂度：$O(n^3)$。

5 矩阵的线性递推

5.1 问题引入

给定一个 n 阶矩阵 A，求出 A^k。所有运算均在模 10^9+7 意义下进行。

数据范围：$n \le 100$，$k \le 10^{10000}$。

5.2 问题分析

题目来源：BZOJ 4162, shlw loves matrix II。

直接矩阵快速幂复杂度是 $O(n^3 \log k)$ 的。而在这道题当中，k 高达 10^{10000}，显然矩阵快速幂并不够快。

定义 5.1 (特征多项式)**.** 矩阵 A 的特征多项式定义为：

$$p_A(\lambda) = |A - \lambda I|$$

定理 5.1 (Cayley-Hamilton 定理)**.**

$$p_A(A) = a_0 + a_1 A + a_2 A^2 + \cdots + a_n A^n = 0$$

我们先求出 A 的特征多项式以及 $A^i (i \le n)$。那么利用 Cayley-Hamilton 定理，我们可以将 A^k 表示为 $A^i (i \le n)$ 的线性组合。具体的系数就是 $x^k \mod p_A(x)$。

$x^k \mod p_A(x)$ 可以利用多项式快速幂在 $O(n^2 \log)$ 的复杂度内计算得到。现在我们来思考一下，在模质数意义下，如何求解特征多项式。

5.2.1 基于维护多项式的解法

将矩阵 A 改造为：

$$A' = \begin{bmatrix} a_{1,1}-x & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2}-x & a_{2,3} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3}-x & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,n}-x \end{bmatrix}$$

然后对矩阵在模 x^{n+1} 意义下做高斯消元，最后把对角线上多项式乘起来就是答案。

由于需要使用多项式乘法，复杂度为 $O(n^5)$。

5.2.2 基于拉格朗日插值的解法

我们对 λ 分别带入 $0, 1, 2, \cdots n$，然后算出 $|A - \lambda I|$。最后根据点值使用拉格朗日插值求出答案。复杂度为 $O(n^4)$。

5.2.3 基于上海森堡矩阵的解法

定义 5.2 (上海森堡矩阵)**.** 如果一个 n 阶矩阵 A 满足：$\forall i > j+1, a_{i,j} = 0$，那么 A 是一个上海森堡矩阵。

假如说我们能将 A 通过如下形式变为一个上海森堡矩阵 A'：

$$A' = P_1 P_2 \cdots P_k A P_k^{-1} \cdots P_2^{-1} P_1^{-1}$$

其中 A_i 为初等变换的矩阵。那么就有：$|A - \lambda I| = |A' - \lambda I|$。因为：

$$\begin{aligned}
|A - \lambda I| &= |P_1 P_2 \cdots P_k (A - \lambda I) P_k^{-1} \cdots P_2^{-1} P_1^{-1}| \\
&= |P_1 P_2 \cdots P_k A P_k^{-1} \cdots P_2^{-1} P_1^{-1} - \lambda P_1 P_2 \cdots P_k P_k^{-1} \cdots P_2^{-1} P_1^{-1}| \\
&= |A' - \lambda I|
\end{aligned}$$

我们考察左乘初等变换的矩阵，发现：将第 i 行加上第 j 行的 k 倍的逆矩阵，如果右乘到 A 上，恰好就是将第 j 列减去第 i 列的 k 倍。

也就是说对 A 的一次行列变换可以被描述为：将第 i 行加上去第 j 行的 k 倍，并且将第 j 列减去第 i 列的 k 倍。那么我们从 2 到 n 枚举 i，每次用第 i 行去消所有 $> i$ 行的第 $i-1$ 列，就可以将 A 变成 A'。

而上海森堡矩阵求行列式是可以使用 $O(n^2)$ 的 dp 来解决的。我们仍然带入 $1 \sim n$ 的点值进行拉格朗日插值。于是我们就可以在 $O(n^3)$ 的时间复杂度内解得特征多项式。

5.2.4 基于 Berlekamp-Massey 算法的解法

Berlekamp-Massey 算法能在 $O(n^2)$ 的时间复杂度内从长为 n 的序列中找到最短递推式。

我们可以直接对矩阵套用 Berlekamp-Massey 算法，但是由于矩阵的加法是 n^2 的，总复杂度就会变为 $O(n^4)$。我们可以随机选择一个行向量 a 和一个列向量 b，然后对 aA^kb 进行递推，这样求递推式的复杂度就为 $O(n^3)$ 了，瓶颈在于求向量乘矩阵。由 Schwartz-Zippel 引理，这样求出正确递推式的概率为 $1 - \frac{2n}{p}$。

这种做法求出的最短递推式长度实际上是矩阵的不同特征值个数。

5.3 例题

给定一张 n 个点，m 条边的有向图，边有边权。第 i 个点的权值为 a_i。每一时刻所有点的权值更新为 $a'_u = \sum_{(v,u)\in E} a_v w(v, u)$。问 k 时刻后所有点的权值。答案对 $10^9 + 7$ 取模。

数据范围：$n \le 5000, k \le 10^{18}$。

5.4 问题分析

暴力矩阵快速幂复杂度为 $O(n^3 \log)$。如果使用上海森堡来计算特征多项式，预处理时仍然有一个 $O(n^3)$，无法接受。

我们考虑使用 Berlekamp-Massey 算法。这个时候题目实际上已经给出了列向量 a，我们只需要再随机找一个行向量 b 相乘就行。预处理前 $n+1$ 时刻的所有 a，然后乘上 b 用于进行 Berlekamp-Massey。最后用多项式快速幂求出线性组合。

6 特征值与特征向量

定义 6.1 (特征值与特征向量)**.** 我们称，如果有非零 v，使得

$$Av = \lambda v$$

那么 λ 称为矩阵的特征值，v 则称为特征向量。

我们将式子变为：$(A - \lambda I)v = 0$。也就是说 $A - \lambda I$ 要线性相关，即 $|A - \lambda I| = 0$。因此 $p_A(\lambda) = 0$，也就是 λ 为特征多项式的根。

6.1 矩阵的对角化

如果一个 n 阶矩阵的所有特征值都互不相同，那么显然其有 n 个线性无关的特征向量。我们将所有特征向量 v_i 拼接起来，并把 λ_i 放到对角线上：$v_{i,j} = (v_i)_j$，$b_{i,j} = [i = j]\lambda_j$。

那么由 $Av_i = \lambda_i v_i$，得到 $AV = VB \Leftrightarrow V^{-1}AV = B$，其中 B 只有对角线上有值。

我们又发现：

$$A^k = P(P^{-1}AP)^k P^{-1}$$

也就是说，通过矩阵对角化，我们将求矩阵的幂转化为了矩阵乘法和对角线上值的幂。

6.2 例题 1

有 n 个编号为 $1, 2, \ldots, n$ 的厨师为国王准备菜肴, 第 i 个厨师的能力值为 i。最初第 i 个厨师的菜肴有美味值 a_i，其中 $|a_i| \le i$。每个厨师有一个允许他复制的厨师名单，国王确保厨师 i 只能从技能更强的厨师那里复制。

在每一天，厨师会进行两个阶段的操作来改变菜肴的美味值：

1. 可以选择令他的菜肴的美味值乘上他的能力值，也可以不乘。

2. 在所有厨师进行完操作 1 后，对于所有在第 i 个厨师的列表中的厨师 j，厨师 i 可以选择将自己的菜肴的美味值加上厨师 j 的菜肴的美味值或者不加。

所有的厨师都尽量令自己菜肴的美味值最大化。

最后有 q 个操作，每种操作是以下两种中的一个：

1. 1 k l r 询问 k 天后 l 到 r 的厨师的菜肴的美味值之和。

2. 2 i x 将第 i 个厨师菜肴初始的美味值 a_i 加上 x，其中 $x > 0$。

$1 \le n \le 300, 1 \le q \le 2 \times 10^5, k \le 10^3, -i \le a_i \le i$。

6.3 问题分析

题目来源：CF1540E, Tasty Dishes。

首先厨师一定是把列表中的所有厨师 $a_j > 0$ 的全部加一遍。由于每个人 $a_i > 0$ 的时间不一样，因此我们考虑对于每个人单独考虑其贡献。我们假设第 i 个人最早 $a_i > 0$ 的时间为 d_i。容易发现 d_i 只与 $a_i > 0$ 的集合有关，并且至多改变 $O(n)$ 次。

我们令 $\vec{e_i}$ 为只有第 i 位为 1 的向量，假设转移矩阵为 T 那么问题变为求

$$\sum_{d_i \le k} T^{k-d_i}\vec{e_i}a_i + \sum_{d_i > k} \vec{e_i}a_i$$

注意到特征向量满足 $Tv_i = \lambda_i v_i$。我们将 e_i 拆为特征向量的线性组合，这个可以 $O(n^3)$ 预处理。然后对于每个特征向量，求出它对于询问 $l \sim r$ 的贡献系数。于是我们现在只需要维护每个 $\lambda_i^k v_i$ 的系数。

我们可以使用 n 个树状数组来维护 $d_i \le k$ 的 e_i 对于 $\lambda_i^k v_i$ 的贡献系数和。于是最后的复杂度为 $O(n^3 + qn\log n)$。

6.4 例题 2

现在有一个随机变量 x。对于 $1 \le i \le n$，x 有 $a_i \times 10^{-9}$ 的概率为 i。

每一时刻 x 有 $\frac{x}{n}$ 的概率减少 1，有 $\frac{n-x}{n}$ 的概率增加 1。问最后 $x = i$ 的概率。答案对 998244353 取模。

$n \le 10^5$。

6.5 问题分析

题目来源：ARC133F, Random Transition。

我们先写出 $n+1$ 阶转移矩阵：

$$A=\begin{bmatrix}0 & \frac{1}{n} & 0 & \cdots & 0 & 0\\ 1 & 0 & \frac{2}{n} & \cdots & 0 & 0\\ 0 & \frac{n-1}{n} & 0 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \cdots & \vdots & \\ 0 & 0 & 0 & \cdots & 0 & 1\\ 0 & 0 & 0 & \cdots & \frac{1}{n} & 0\end{bmatrix}$$

令 A 的第 $i+1$ 小的特征值为 λ_i，对应的特征向量为 v_i。我们可以发现：$\lambda_i=\frac{2i}{n}-1$。而特征向量 v_i 为多项式 $(1+x)^i(1-x)^{n-i}$ 的系数，即 $(v_i)_j=[x^j](1+x)^i(1-x)^{n-i}$。并且如果将 v_i 拼接为 P，那么 P^{-1} 的第 $i+1$ 列为 $n^{-2}v_{n-i}$。

对于 $b=Pa$，有

$$b_i=\sum_{j=0}^{n}a_j[x^i](1+x)^j(1-x)^{n-j}$$

那么，我们只需要求出

$$\sum_{i=0}^{n}a_i(1+x)^i(1-x)^{n-i}$$

直接分治 NTT 可以在 $O(n\log^2 n)$ 的时间内求出。

但实际上，我们可以做得更优。令 $t=1+x$，问题变为求 $\sum_{i=0}^{n}a_it^i(2-t)^{n-i}$。发现 a_i 对于 $[t^j]$ 的贡献为一个组合数，可以拆开以后利用 NTT 求出。然后我们需要利用 $f(t-1)$ 还原得到 $f(x)$，直接将 $f(t-1)$ 展开，贡献仍为组合数，不再赘述。

于是，我们在 $O(n\log n)$ 的时间复杂度内计算出了 $b=Pa$。对于 $b=P^{-1}a$ 我们也能类似地计算。而中间计算 A'^k 只需要 n 次快速幂，因此整个问题的复杂度为 $O(n\log n)$。

6.6 小结

在上面两道例题中，我们利用了矩阵的特征值以及特征向量，将原本难以计算的矩阵快速幂转化为了对角线上元素的快速幂，从而降低了复杂度。实际上，第二道题的官方做法并不是矩阵快速幂，而是观察性质后求生成函数。但是在我们无法观察出性质的时候，将转移矩阵的特征向量进行打表后往往能有新的发现。

7 特征值的计算

第 6 节中讨论的情况基本都是能直接求解出特征值的，在本节我们将针对更为一般的情形，介绍三种求数值解的方法以及一种求模意义下解的方法。这些方法也可以帮助我们进行打表的工作。

7.1 迭代幂法

对于 n 阶矩阵 A，如果其绝对值最大的特征值唯一，那么我们可以利用迭代幂法求出 A 绝对值最大的特征值以及对应的特征向量。

我们先随便找一个非零列向量 x，然后重复以下步骤：

1. 计算 $x' = Ax$。
2. 将 x' 缩放使得 x' 的模长为 1。
3. 将 x 赋为 x'。

重复若干步后 x 将会收敛于 $|\lambda|_{max}$ 的 v。

证明. 我们先将 x 分解为特征向量 v_i 的线性组合：$x = \sum_i p_i v_i$。那么迭代 k 次之后，有 $x = \sum_i \lambda_i^k p_i v_i$。那么经过若干次迭代后，其他向量的 λ_i^k 相对于 $|\lambda|_{max}^k$ 可以忽略不计，也就是说 x 里几乎只剩下了最大特征向量。

□

类似地，将整个过程反过来可以求出 $|\lambda|$ 最小的特征值以及特征向量。

7.2 Jacobi 方法

定义 7.1 (Givens 旋转矩阵)**.** *Givens* 旋转矩阵 $R(p, q, \theta)$ 定义为：

$$R(p,q,\theta)_{i,j} = \begin{cases} 1, & i = j, i \neq p, i \neq q \\ \cos\theta, & i = j = p 或 i = j = q \\ \sin\theta, & i = p, j = q \\ -\sin\theta, & i = q, j = p \end{cases}$$

对于实对称矩阵 A（$A = A^T$），求出 $B = R(p,q,\theta)AR^T(p,q,\theta)$，那么有：

$$\begin{cases} b_{i,j} = a_{i,j}, & i,j \neq p,q \\ b_{p,i} = b_{i,p} = \cos\theta a_{p,i} + \sin\theta a_{q,i}, & i \neq p,q \\ b_{q,i} = b_{i,q} = \cos\theta a_{q,i} - \sin\theta a_{p,i}, & i \neq p,q \\ b_{p,q} = b_{q,p} = \sin\theta\cos\theta(a_{q,q} - a_{p,p}) + (\cos^2\theta - \sin^2\theta)a_{p,q}\theta \\ b_{p,p} = \sin^2\theta a_{p,p} + \cos^2\theta a_{q,q} - 2sin\theta\cos\theta a_{p,q} \\ b_{q,q} = \cos^2\theta a_{p,p} + \sin^2\theta a_{q,q} + 2sin\theta\cos\theta a_{p,q} \end{cases}$$

容易发现有 $\sum_{i,j} b_{i,j}^2 = \sum_{i,j} a_{i,j}^2$。

我们注意到 $b_{p,q} = b_{q,p} = \frac{1}{2}\sin 2\theta(a_{q,q} - a_{p,p}) + \cos\theta a_{p,q}$。那么令 $\tan 2\theta = \frac{2a_{p,q}}{a_{q,q}-a_{p,p}}$，就有 $b_{p,q} = b_{q,p} = 0$。此时发现 B 的对角线元素的平方和相比 A 增加了 $2a_{p,q}^2$。

我们每次都找到 $|a_{p,q}|$ 最大的 p,q 进行操作，那么在经过足够次数的迭代后，A 的非对角线元素会趋近于 0。此时 A 对角线上元素的就是特征值，而之前操作过的 $R(p,q,\theta)$ 按顺序依次相乘就可以得到特征向量。

令 $\mathtt{T}(A)$ 表示非对角线元素的平方和。那么显然会有 $\mathtt{T}(B) \le (1 - \frac{2}{n(n-1)})\mathtt{T}(A)$。为了使 $\mathtt{T}(B) < \epsilon$，则要进行至少 $\frac{\ln\mathtt{T}(A) - \ln\epsilon}{\ln(n(n-1)) - \ln(n(n-1)-2)} \approx \frac{n^2}{2}(\ln\mathtt{T}(A) - \ln\epsilon)$ 次操作。如果使用数据结构维护平方最大值，则可以实现 $O(n^3(\ln\mathtt{T}(A) - \ln\epsilon)\log n)$ 的复杂度。而实际上，我们发现，如果我们对于每行维护一个最大值所在位置，那么每次操作期望只会改变 $O(1)$ 行的最大值，于是可以实现 $O(n^3(\ln\mathtt{T}(A) - \ln\epsilon))$ 的复杂度。

7.3 QR 分解

定理 7.1 (QR 分解)**.** 任何实数非奇异矩阵 A 都存在 QR 分解，即：

$$A = QR$$

其中若限定 R 对角线元素为正，则该分解是唯一的。这里 Q 为正交矩阵，也就是 $Q^TQ = I$，而 R 为上三角矩阵。

7.4 Householder 变换

Householder 变换可以用于求解矩阵的 QR 分解。其具体流程为：

1. 令 x 为矩阵 A 第一列的向量，令 e_1 为 n 维列向量 $[1,0,0,\cdots,0]^T$。

2. 计算 $v = x - |x|e_1$，其中 $|x|$ 为 x 的模长。

3. 计算 $\omega = \frac{v}{|v|}$。

4. 计算 $Q = I - 2\omega\omega^T$。

容易发现这样计算出的 Q 满足 QA 只有第一列只有第一行处有值。这启发我们像高斯消元那样递归下去，得到最终的 R。也就是 $R = Q_n \cdots Q_2Q_1A$，那么我们便可以得到：$A = Q_1^TQ_2^T \cdots Q_n^TR$，也就是 $Q = Q_1^TQ_2^T \cdots Q_n^T$。

7.5 QR 迭代法

对于 n 阶非奇异矩阵 A，进行 QR 分解：$A_1 = Q_1R_1$。然后按照如下公式迭代：

$$A_{k+1} = R_kQ_k = Q_k^TA_kQ_k = Q_{k+1}R_{k+1}$$

那么若 A 的特征值的绝对值互不相同，则有 $\lim_{k\to\infty} A_k = T$。其中 T 是一个上三角矩阵。

而又有 $T = Q_k^T \cdots Q_2^TQ_1^TA_1Q_1Q_2 \cdots Q_k$，于是得到 $|A| = |T|$。而上三角矩阵的特征值就是对角线上元素。得到特征值后可以通过高斯消元解方程解出特征向量。

由于篇幅有限，此处不给出此方法收敛的证明，感兴趣的读者可以下来自己阅读参考文献。

7.6 多项式求根

在实数域上由于精度有限，使用多项式求根来求出特征值往往误差会比较大，因此一般都不会使用这种方法。然而如果是在模质数意义下求特征值时就可以采用这种方法。

在模 p 意义下对多项式进行分解有许多种做法，这里作者提供一种：每次将 $F(x)$ 的系数随机平移 d，求出 $\gcd(F(x+d), x^{\frac{p-1}{2}})$。显然，如果 $x - w | x^{\frac{p-1}{2}} - 1$，那么 w 在模 p 意义下有二次剩余。那么每次求 $\gcd(F(x+d), x^{\frac{p-1}{2}})$ 会期望得到 $F(x)$ 一半的根，可以将 $F(x)$ 分为两部分分别递归求解。

那么我们只需要使用 3.1.3 中提到的算法求出特征多项式，然后求出所有的根就可以得到特征值。再跑 n 次高斯消元就可以解出特征向量。

8 总结

本文探讨了矩阵的行列式、特征方程、特征值以及特征向量在信息学竞赛中的应用。信息学中的动态规划（即 dp）与矩阵也有着非常紧密的联系，许多 dp 都可以通过矩阵快速幂进行优化。在一些计数题中，如果难以优化 dp，那么可以尝试列出转移矩阵并且求出特征向量后打表观察规律，或许就会有不小的收获。

矩阵特征值与特征向量的数值解计算还有其他的许多算法，碍于笔者的水平有限，并没有进行进一步介绍。感兴趣的读者可以自行研究一下参考资料中的内容。

9 致谢

感谢中国计算机学会提供学习和交流的平台。
感谢父母对我的关心和支持。
感谢林鸿老师、蔺洋老师和叶诗富老师对我的培养。
感谢所有伴我走过 OI 之路的人。

参考文献

[1] Characteristic Polynomial, https://en.wikipedia.org/wiki/Characteristic_polynomial

[2] Berlekamp-Massey algorithm, https://en.wikipedia.org/wiki/Berlekamp%E2%80%93Massey_algorithm

[3] Eigenvalues and eigenvectors, https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

[4] QR decomposition, https://en.wikipedia.org/wiki/QR_decomposition

[5] Householder transformation, https://en.wikipedia.org/wiki/Householder_transformation

[6] 基于 QR 分解的特征值迭代法, https://blog.csdn.net/weixin_44246009/article/details/115263946

[7] Jacobi eigenvalue algorithm, https://en.wikipedia.org/wiki/Jacobi_eigenvalue_algorithm

[8] 非对称特征值问题的计算方法, https://www.cnblogs.com/Bluemultipl/p/15913516.html

[9] 经典 Jacobi 方法用于求解矩阵特征值, https://zhuanlan.zhihu.com/p/262879394