

矩阵树定理

Orange

2018 年 5 月 26 日

1 摘要

矩阵树定理 (Matrix Tree Theorem) 是用于解决有标号图的生成树 (树形图) 计数问题的有力工具。本文以知识的广度为侧重点, 从矩阵树定理的数学基础提起, 并对其进行了简单的证明, 最后给出矩阵树定理的简单扩展及应用。

关键词: 矩阵树定理 行列式

2 数学基础

2.1 矩阵

2.1.1 定义

数学中, 矩阵被定义为一个矩形的二维数组。一般地, 我们用如下方式表示一个 n 行 m 列的矩阵:

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

特别地, 如果一个矩阵的行数和列数相等, 即 $n = m$, 那么我们称这个矩阵为方阵。

2.1.2 矩阵的转置

有一 $n \times m$ 的矩阵 A , 我们定义其转置为一个 $m \times n$ 的矩阵, 且它的转置的元素 $a'_{i,j}$ 满足:

$$a'_{i,j} = a_{j,i}$$

我们记矩阵 A 的转置为 A^T 。

2.1.3 矩阵乘法

设 $n \times k$ 的矩阵 A 和 $k \times m$ 的矩阵 B ，我们定义 $A \times B$ 的结果为一个 $n \times m$ 的矩阵，若将其记为 C ，则对其中的元素 $c_{i,j}$ 有：

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

矩阵乘法不满足交换律，但是满足结合律。

2.1.4 高斯消元

高斯消元指一种用于计算增广矩阵 (augmented matrix) 所表示的方程组的解的算法，具体的内容不再详细阐述，在此，只给出高斯消元的一些简单变形。

模意义下的高斯消元

一般的高斯消元都用于解决实数域上的线性方程组问题，但有时需要求出模意义下的解，这个怎么解决呢？事实上，我们只需要把每一步的操作（即矩阵初等变换）变成模意义下的就可以了。如何证明它的正确性？这里可以感性地理解下：

对于增广矩阵的一行，可以看作：

$$ax + by + cz = d$$

设模数为 m 。在模意义下，上式表示：

$$((a \bmod m)(x \bmod m) + (b \bmod m)(y \bmod m) + (c \bmod m)(z \bmod m)) \bmod m = d \bmod m$$

令 a 加上 a' ， b 加上 b' ， c 加上 c' ， d 加上 d' ， a', b', c', d' 表示矩阵中另一行的数。以 a 为例，将会变成以下形式：

$$((a + a') \bmod m)(x \bmod m)$$

由于在模意义下满足乘法分配律，因此可以将 a, b, c, d 和 a', b', c', d' 分开（为了方便，写成同余式）：

$$(a + a')x \equiv ax + a'x \pmod{m}$$

最后可得：

$$(ax + by + cz) + (a'x + b'y + c'z) \equiv d + d' \pmod{m}$$

由于 $ax + by + cz \equiv d \pmod{m}$ 且 $a'x + b'y + c'z \equiv d' \pmod{m}$ ，因此模意义下的矩阵初等变换是正确的。

使用辗转相除法的高斯消元

在矩阵初等变换时，为了消元，我们需要为某一行乘上一个系数，在实数域下我们利用除法来得到这个系数。类似地，在模意义下，我们可以利用逆元来得到这个系数。由于这只是进行了一次矩阵初等变换，因此这么做是正确的。

但并不是所有模数都是质数。当模数不是质数，或者我们计算的答案为大整数，需要避免浮点误差时，逆元和除法就不能用了。但我们可以利用辗转相除法的思想，通过两行互相消元来达到目的。具体的操作请参考下面的代码，其时间复杂度为 $O(n^3 \log n)$ 。

```
1 int main() { return 0; }
```

2.2 行列式

2.2.1 定义

记：

$$D = \begin{vmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{vmatrix}$$

称为 n 阶行列式。

不同于矩阵，一个行列式表示一个数，具体的运算法则我们将在后文提及。

对一个 $n \times n$ 的方阵 A ，记：

$$\det(A) = \begin{vmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{vmatrix}$$

其中 \det 是一个函数，表示求矩阵 A 相应的行列式。

为了方便，我们记 D_n 表示一个一般的 n 阶行列式，即：

$$D_n = \begin{vmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{vmatrix}$$

2.2.2 排列与逆序数

我们称由 $1 \sim n$ 组成的一个有序数组 $a_{1 \sim n}$ 为一个 n 级排列。显然，共有 $n!$ 个不同的 n 级排列。

在一个 n 级排列中，若某两个数的前后位置与大小顺序不同，即大的在小的前面，则称这两个数构成一个逆序对。称排列 a 中逆序对的总数为这个排列的逆序数，记为 $N(a)$ 。

如果一个排列的逆序数为奇数，我们称它为奇排列；如果一个排列的逆序数为偶数，我们称它为偶排列。有一个不是很显然的结论：对于 $n!$ 个不同的 n 级排列，奇排列和偶排列的个数分别为 $\frac{n!}{2}$ ，证明略。

2.2.3 对角线

我们称从左上角 $(a_{1,1})$ 到右下角 $(a_{n,n})$ 为主对角线；称从右上角 $(a_{1,n})$ 到左下角 $(a_{n,1})$ 为次对角线。主对角线和次对角线各仅有一条。

2.2.4 计算行列式

我们定义 n 阶行列式的运算结果为:

$$D_n = \sum (-1)^{N(j_1 j_2 \cdots j_n)} (a_{1,j_1} \cdot a_{2,j_2} \cdots a_{n,j_n})$$

其中 $j_1 j_2 \cdots j_n$ 代表 n 的一个排列, 而求和表示遍历所有的 n 级排列。

根据定义, 我们有如下较为简单, 但极为重要的结论:

1. 下三角行列式的值等于其主对角线各元素的乘积。

$$\begin{vmatrix} a_{1,1} & 0 & 0 & \cdots & 0 \\ a_{2,1} & a_{2,2} & 0 & \cdots & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,n} \end{vmatrix} = \prod_{i=1}^n a_{i,i}$$

2. 上三角行列式的值等于其主对角线各元素的乘积。

$$\begin{vmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ 0 & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ 0 & 0 & a_{3,3} & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n,n} \end{vmatrix} = \prod_{i=1}^n a_{i,i}$$

3. 对角行列式的值等于其主对角线各元素的乘积。

$$\begin{vmatrix} a_{1,1} & 0 & 0 & \cdots & 0 \\ 0 & a_{2,2} & 0 & \cdots & 0 \\ 0 & 0 & a_{3,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n,n} \end{vmatrix} = \prod_{i=1}^n a_{i,i}$$

类似地, 对于除次对角线外的元素均为 0 的行列式, 它的值为:

$$(-1)^{\frac{n(n-1)}{2}} \prod_{i=1}^n a_{i,n+1-i}$$

另外, 对于一般的行列式 D_n , 它也可以用如下的式子计算:

$$D = \sum (-1)^{N(i_1 i_2 \cdots i_n) + N(j_1 j_2 \cdots j_n)} (a_{i_1, j_1} a_{i_2, j_2} \cdots a_{i_n, j_n})$$

其中 $i_1 i_2 \cdots i_n$ 为一个 n 级排列, $j_1 j_2 \cdots j_n$ 为所有的 n 级排列。

练习

e.g. 1

$$\begin{vmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & n-1 \\ n & 0 & 0 & \cdots & 0 \end{vmatrix}$$
$$=(-1)^{N(23\cdots n1)}(1 \times 2 \times \cdots \times n)$$
$$=(-1)^{n-1}n!$$

e.g. 2

$$\begin{vmatrix} a_1 & b_1 & c_1 & d_1 & e_1 \\ a_2 & b_2 & c_2 & d_2 & e_2 \\ a_3 & b_3 & 0 & 0 & 0 \\ a_4 & b_4 & 0 & 0 & 0 \\ a_4 & b_5 & 0 & 0 & 0 \end{vmatrix} = 0$$

对于行列式的每一项，无法避免至少有一个数为 0，所以行列式为 0。

2.2.5 行列式初等变换

我们可以利用行列式的定义直接计算行列式，但是这么做的时间复杂度是指数级的，没有应用价值。如果我们有一个对角行列式，那么我们就能在线性时间复杂度内计算它。

将一个一般的行列式转变成等效的对角行列式的方法就是利用行列式的性质进行行列式初等变换。

性质 1. $D_n = D_n^T$ ，其中 D_n^T 为 D_n 的转置。

性质 2. 互换行列式的两行或两列，行列式的值变号。

推论 1. 若 D_n 的两行或两列完全相同，则 $D_n = 0$ 。

性质 3. 用数 k 乘以行列式的某一行或某一列，等于 k 乘以行列式。

推论 2. D 中某一行或某一列中所有元素的因子可以提到行列式的外面。

推论 3. 若 D 的某两行或某两列的对应元素成比例，则 $D = 0$ 。

性质 4. 行列式 D 的某一行或某一列的所有元素都乘以数 k 加到另一行或另一列（行对行，列对列）的相应元素上，行列式的值不变。

利用性质对行列式进行“高斯消元”

观察性质 4，可以发现，我们完全可以仿照高斯消元处理行列式的方法，将行列式变成一个上三角矩阵。可以这样做的原因是 矩阵初等变换与行列式初等变换是类似的。借此我们得到了一个时间复杂度为 $O(n^3)$ 的计算行列式的值的算法：

```
1 int main() { return 0; }
```

2.2.6 子式

在一个矩阵或行列式中，从中选取 k 行 k 列，这 k 行 k 列相交的元素个数一定为 k^2 个，我们把这 k^2 个元素按照相对位置排列成一个新的行列式，叫作原矩阵或行列式的一个 k 阶子式。

主子式

对于行列式 D_n ，我们称选择 $r_{i_1}r_{i_2}\cdots r_{i_k}$ 行和 $r_{i_1}r_{i_2}\cdots r_{i_k}$ 列构成的子式为行列式 D_n 的一个 k 阶主子式。即对主子式而言，行和列选择的标号是相同的。

主子式的一个重要特征是：对于在被选行（列）的主对角线上的元素，它们均存在于主子式中。

余子式

对于行列式 D_n ，划去第 i 行与第 j 列元素后，所留下来的 $n-1$ 阶行列式叫做余子式，记作 $M_{i,j}$ ；也可被称为 $a_{i,j}$ 的余子式。

代数余子式

我们记 $A_{i,j}$ 表示 $a_{i,j}$ 的代数余子式， $A_{i,j}$ 满足：

$$A_{i,j} = (-1)^{i+j} M_{i,j}$$

2.2.7 Binet-Cauchy 公式

设 A 是一个 $n \times m$ 的矩阵， B 是一个 $m \times n$ 的矩阵，有：

$$\det(A \times B) = \begin{cases} 0 & n > m \\ \sum_{1 \leq k_1 < k_2 < \cdots < k_n \leq n} \det(A_{k_1 k_2 \cdots k_n}) \cdot \det(B_{k_1 k_2 \cdots k_n}) & n \leq m \end{cases}$$

其中 $A_{k_1 k_2 \cdots k_n}$ 表示 A 的第 k_1, k_2, \cdots, k_n 列所形成的子式， $B_{k_1 k_2 \cdots k_n}$ 表示 B 的第 k_1, k_2, \cdots, k_n 行所形成的子式。这个公式被称为比内-柯西（Binet-Cauchy）公式。

对 Binet-Cauchy 公式的证明超出了我们的范围，但是这个公式是证明矩阵树定理不可少的公式，有兴趣的读者可以自行搜集资料。

3 矩阵树定理

如不加说明，我们认为我们的图有 n 个点， m 条有向边，且不存在自环，但可以有重边。并且我们规定 $n \geq 2$ ，因为这种特殊情况会影响我们的推导，再者，此时答案显然为 1。

3.1 三个重要矩阵

3.1.1 图 G 的入度矩阵 D

我们定义图 G 的入度矩阵 D 是一个 $n \times n$ 的矩阵，满足：

$$\begin{cases} D_{i,j} = 0 & (i \neq j) \\ D_{i,i} = d_i \end{cases}$$

其中 d_i 表示点 i 的度数。

3.1.2 图 G 的邻接矩阵 A

图 G 的邻接矩阵 A 也是一个 $n \times n$ 的矩阵，满足：

$$\begin{cases} A_{i,i} = 0 \\ A_{i,j} = a_{i,j} & (i \neq j) \end{cases}$$

其中 $a_{i,j}$ 表示 i 到 j 的边数。如果保证不存在重边，那么这个矩阵一定是一个仅含有 0 和 1 的矩阵。

3.1.3 图 G 的基尔霍夫矩阵

我们定义图 G 的基尔霍夫 (Kirchhoff) 矩阵 K 为一个 $n \times n$ 的矩阵，满足：

$$K = D - A$$

3.2 定理

对于有向图 G ，它以 i 为根的树形图个数为它的基尔霍夫矩阵 K 去掉第 i 行的主子式，即：

$$\text{Ans} = M_{i,i}$$

其中 $M_{i,i}$ 表示 K 去掉第 i 行和第 i 列的余子式，其表示的内容与去掉第 i 行的主子式相同。以上就是矩阵树定理 (Matrix Tree Theorem) 的全部内容。

3.3 矩阵树定理的证明

3.3.1 特殊情况

基尔霍夫矩阵的行列式

可以证明：对于任意图的基尔霍夫矩阵 K ，有：

$$\det(K) = 0$$

证明. 可以发现，基尔霍夫矩阵的每一行的和都是 0。我们将所有列都加到第一列中，则第一列将全部变成 0，此时易得 $\det(K) = 0$ 。□

当这张图是一个以 i 为根的树形图时

可以证明：如果这张图是一个以 i 为根的树形图，那么有：

$$\det(M_{i,i}) = 1$$

其中 $M_{i,i}$ 表示去掉第 i 行的 $n-1$ 阶主子式。

证明. 我们对结点重新标号，按照它们的深度从小到大标号，显然这并不会改变我们的结果。

设 $M = M_{1,1}$ ，很明显， M 主对角线上的值均为 1，因为每个结点只有唯一的父结点，这使得每个点的入度均为 1（根结点除外）。由于我们按深度排序，因此不存在编号大的结点到编号小的结点的边，这使得 M 是一个上三角矩阵。因此， M 的值为 1。□

当这张图有 $n-1$ 条边且弱连通但不是以 i 为根的树形图时

可以证明：如果这张图有 $n-1$ 条边且弱连通但不是以 i 为根的树形图，那么有：

$$\det(M_{i,i}) = 0$$

证明. 如果一张图为树形图，那么入度为 1 的点将会有 $n-1$ 个，换句话说，入度为 0 的点只有 1 个，即 i 。若将这张树形图的边任意反向（至少对一条边进行反向），那么一定至少有一个除了 i 之外的点的入度变为 0。根据基尔霍夫矩阵的定义，这张图去掉第 i 行的 $n-1$ 阶主子式的值为 0。□

当这张图不连通时

可以证明：如果这张图不连通，那么它的任意一个 $n-1$ 阶主子式均为 0。

证明. 我们不妨设图被分成了两个连通块。现在我们对这张图进行重编号，显然这不会改变我们的结果。我们令第一个连通块中的点的编号始终小于第二个连通块中点的编号，那么它的基尔霍夫矩阵的大致形态为：

$$\begin{vmatrix} a_{1,1} & a_{1,2} & a_{1,3} & 0 & \cdots & 0 & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & 0 & \cdots & 0 & 0 & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{4,4} & \cdots & b_{4,n-2} & b_{4,n-1} & b_{4,n} \\ 0 & 0 & 0 & b_{5,4} & \cdots & b_{5,n-2} & b_{5,n-1} & b_{5,n} \\ 0 & 0 & 0 & b_{6,4} & \cdots & b_{6,n-2} & b_{6,n-1} & b_{6,n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & b_{n,4} & \cdots & b_{n,n-2} & b_{n,n-1} & b_{n,n} \end{vmatrix}$$

由于两个连通块之间不存在任何边，因此在行列式中有两个所有元素均为 0 的区域。

现在我们对行列式进行变换，把所有 a 所在的列加到 a 所在的第一列，同理，把 b 所在的所有列加到 b 所在的第一列，那么这个行列式会有两个全是 0 的列。因此，任何一个 $n-1$ 阶主子式的值均为 0。□

3.3.2 关联矩阵

我们定义图 G 的关联矩阵 B 为一个 $n \times m$ 的矩阵，其中每一行代表一个点，每一列代表一条边。对于每条边（每列），如果这条边是从 a 到 b 的，那么我们令这一列的第 a 行等于 1，第 b 行等于 -1，其余行等于 0。这样，我们得到了一个每一列有一行为 1，一行为 -1，其余行为 0 的矩阵。

我们定义图 G 的转置关联矩阵 B^T 为一个 $m \times n$ 的矩阵，其中每一列代表一个点，每一行代表一条边。注意转置关联矩阵并不是关联矩阵的转置。对于每条边（每行），如果这条边是从 a 到 b 的，那么我们仅令这一行的第 b 列等于 -1，其余行等于 0。显然 B^T 是一个每一行有且仅有一列为 -1，其余行为 0 的矩阵。

我们来研究一下 BB^T 的结果，显然这是一个 $n \times n$ 的矩阵，并且不难发现其结果的第 i 行第 j 列的元素为 B 的第 i 行与 B^T 的第 j 列的点积。我们来举几个例子：

$$\begin{aligned} B_i^T & (0, \quad 0, \quad 0, \quad -1, \quad \dots) \\ B_i & (0, \quad 1, \quad 0, \quad -1, \quad \dots) \\ B_j^T & (0, \quad -1, \quad -1, \quad 0, \quad \dots) \end{aligned}$$

其中 B_i 代表 B 的第 i 行的 m 个元素组成的向量， B_j^T 代表 B^T 的第 j 列的 m 个元素组成的向量。

对于 B_i ，出现一个 1 就代表这一列对应的边是 i 的出边，出现一个 -1 就代表它是 i 的入边；对于 B_j^T ，出现一个 -1 就代表这一行对应的边时 j 的入边。以上面的例子为例，第一列对应的边要么与 i 和 j 都无关，要么是 j 的一条出边；第二列对应的边是从 i 到 j 的边；第三列对应的边是 j 的一条入边，不以 i 为端点；第四列对应的边是 i 的一条入边，可能是 j 的出边。

当 $i = j$ 时，得到的结果是主对角线上的元素，仅当一条边为该点的入边时，它才对这个元素有 1 的贡献，否则不难验证它对这个元素的贡献为 0。

当 $i \neq j$ 时，得到的结果自然不是主对角线上的元素。当 i 到 j 存在一条编号为 k 的边时， B_i 的第 k 个元素为 1， B_j^T 的第 k 个元素为 -1，因此这条边对第 i 行第 j 列这个元素的贡献为 -1。其它情况下，不难验证这条边对这个元素的贡献为 0。

综合以上概念，我们可以得出一个十分重要的结论：

$$BB^T = K$$

也就是说，我们可以通过研究 G 的关联矩阵和转置关联矩阵来研究 K 。

3.3.3 应用 Binet-Cauchy 公式

稍微推算一下即可发现， K 去掉第 r 行的 $n-1$ 阶主子式等于去掉第 r 行的 B （记作 B_r ）与去掉第 r 列的 B^T （记作 B_r^T ）的乘积，即：

$$M_{r,r} = B_r B_r^T$$

现在 B_r 为一个 $(n-1) \times m$ 的矩阵， B_r^T 为一个 $m \times (n-1)$ 的矩阵，并且不要忘了 B_r 的每一列或 B_r^T 的每一行都分别代表一条边。当 $m < n-1$ 时，按常识， $M_{r,r}$ 的值应该为 0，那么 $\det(BB_r)$ 的值是 0 吗？Binet-Cauchy 公式已经解决了这个问题：一个 $N \times M$ 的矩阵与一个 $M \times N$ 的矩阵相乘，若 $M < N$ ，则它们的乘积的行列式为 0。

当 $m \geq n-1$ 时，我们仍然应用 Binet-Cauchy 公式去解决它。显然，总共有 C_m^{n-1} 个求和的项，我们考虑每一项代表了什么。显然，我们可以看作在 m 条边中选择其中 $n-1$ 条边来构成一个基尔霍夫矩阵的 $n-1$ 阶主子式。

至此，矩阵树定理的证明已经基本完成了。我们考虑了所有由 $n-1$ 条边组成的集合，尝试它们能否构成一个以 r 为根的树形图。若它们能够构成树形图，显然这 $n-1$ 条边对应的基尔霍夫矩阵去掉第 r 行的 $n-1$ 阶主子式为 1，对最终答案有 1 的贡献；否则这 $n-1$ 条边构成的图要么不连通（即忽略边的方向后存在环），要么不构成树形图结构，对答案的贡献为 0。

证毕。

3.4 应用举例

3.4.1 无向图的生成树计数

由于是无向图，我们只需要将每条无向边拆成两条有向边即可。显然，我们以任意一个结点为根求出的树形图数量就是这张图的生成树数量。

3.4.2 无向完全图的生成树计数

由 Cayley 定理可知，有标号无向完全图的生成树计数的答案为：

$$n^{n-2}$$

过去，我们用 Prüfer 序列的性质推导出这个结果。现在我们可以用矩阵树定理来解决这个问题。

证明. 无向完全图的基尔霍夫矩阵为：

$$\begin{vmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & -1 & \cdots & -1 \\ -1 & -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & n-1 \end{vmatrix}$$

其 $n-1$ 阶主子式的大致形态与它相同。发现，现在每一列的和均为 1，我们将所有其余行都加到第一行，则第一行的所有值都变成了 1：

$$\begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ -1 & n-1 & -1 & \cdots & -1 \\ -1 & -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & n-1 \end{vmatrix}$$

我们再将其余行全部加上第一行，则其余行只有主对角线上有值 n ：

$$\begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & n & 0 & \cdots & 0 \\ 0 & 0 & n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & n \end{vmatrix}$$

显然其值为 n^{n-2} 。

□

4 例题

脱离了理论的矩阵树定理是比较容易的，因此这里只选出了最基本的题目和用了一些技巧的题目。

4.1 [CQOI 2018] 社交网络

4.1.1 题目大意

给定一张 n 个点 m 条边的有向连通图，求出它以 1 为根的树形图个数。答案对 $10^4 + 7$ 取模。

4.1.2 解决方法

直接使用矩阵树定理求行列式即可。不过这道题需要取模，因此我们需要保证我们求行列式的方法在模意义下仍然正确。

可以发现，上三角行列式的结果是主对角线上元素的乘积，因此最终的结果等于主对角线上的元素在模意义下的乘积。由于模意义下的高斯消元是正确的，因此直接在模意义下进行行列式初等变换是正确的。再次提醒，行列式交换两行时需要变号，因此必须记录交换次数的奇偶性。

可以验证 $10^4 + 7$ 是一个质数，因此我们可以求出它的逆元，用乘以逆元的方式代替除法即可。