线性代数初步

Linear Algebra in OI (Simple Edition)

SGColin

February 18, 2021

石家庄二中 信息学竞赛集训

Preface

- 高义雄
- 石家庄二中南校区 2017 级（2020 届）
- APIO 2019 铜牌 NOIP 2018 一等奖
- The 2020 ICPC Asia Jinan Regional 金牌

- 内容比较多，节奏会比较快
- 有问题随时提问

Basic

Matrix - Definition

矩阵是一个 m 行 n 列的二维数表。

$m \times n$ 阶**矩阵** $A_{m\times n}$： 行数 m， 列数 n， 元素 $a_{i,j}$（i 行 j 列）

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

当行数和列数相同时，即 $m = n$ 时，称其为 n **阶方阵**。

记号：主对角线（$a_{1,1} \rightarrow a_{n,n}$），副对角线（$a_{1,n} \rightarrow a_{n,1}$）

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix}$$

Matrix - Definition

```
struct Matrix {

    int m, n;

    int a[N][N];

    inline void init(int _m = 0, int _n = 0) {
        m = _m; n = _n;
        memset(a, 0, sizeof(a));
    }

};
```

- 零矩阵 $\mathrm{O}_{m\times n}$： 所有元素全部为 0
- 单位阵 I_n： n 阶方阵，主对角线上为 1 ，其余位置全部为 0 。
- 对角阵 Λ_n： n 阶方阵，记号 $\Lambda_n = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$

$$\Lambda_n = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

矩阵加法（减法）：对应位置元素相加（减）。

只有**行数和列数均相同**的两个矩阵才可以相加（减）。

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

$$A \pm B = \begin{pmatrix} a_{1,1} \pm b_{1,1} & a_{1,2} \pm b_{1,2} & \cdots & a_{1,n} \pm b_{1,n} \\ a_{2,1} \pm b_{2,1} & a_{2,2} \pm b_{2,2} & \cdots & a_{2,n} \pm b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} \pm b_{m,1} & a_{m,2} \pm b_{m,2} & \cdots & a_{m,n} \pm b_{m,n} \end{pmatrix}$$

Matrix - Addition & Subtraction

```
inline Matrix operator + (Matrix B) {
    Matrix res;
    res.init(m, n);
    for (int i = 1; i <= m; ++i)
        for (int j = 1; j <= n; ++j)
            res.a[i][j] = a[i][j] + B.a[i][j];
    return res;
}

inline Matrix operator - (Matrix B) {
    Matrix res;
    res.init(m, n);
    for (int i = 1; i <= m; ++i)
        for (int j = 1; j <= n; ++j)
            res.a[i][j] = a[i][j] - B.a[i][j];
    return res;
}
```

矩阵数乘（标量乘法）：所有位置元素乘以常数 c 。

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

Matrix - Scalar Multiplication

```cpp
inline Matrix operator * (int c) {
    Matrix res;
    res.init(m, n);
    for (int i = 1; i <= m; ++i)
        for (int j = 1; j <= n; ++j)
            res.a[i][j] = c * a[i][j];
    return res;
}
```

定理 设A,B,C是同型的矩阵，λ，μ为数，则

$$A+B=B+A$$

$$(A+B)+C=A+(B+C)$$

$$A_{m\times n}+O_{m\times n}=A=O+A$$

$$A+(-A)=O=(-A)+A$$

$$\lambda(\mu A)=(\lambda\mu)A=\mu(\lambda A)$$

$$(\lambda+\mu)A=\lambda A+\mu A$$

$$\lambda(A+B)=\lambda A+\lambda B$$

矩阵乘法： 这里我们不加证明的给出两个矩阵乘法的结果形式。

注 两矩阵可以做乘法的前提是 A 的列数 = B 的行数。

$$AB=\begin{pmatrix} a_{11} & \cdots & a_{1s} \\ \vdots & & \vdots \\ \boxed{a_{i1} \quad \cdots \quad a_{is}} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{ms} \end{pmatrix} \begin{pmatrix} b_{11} & \cdots & \boxed{\begin{matrix} b_{1j} \\ \vdots \\ b_{sj} \end{matrix}} & \cdots & b_{1n} \\ \vdots & & & & \vdots \\ b_{s1} & \cdots & & \cdots & b_{sn} \end{pmatrix} \triangleq \begin{pmatrix} \ddots & & \iddots \\ & c_{ij} & \\ \iddots & & \ddots \end{pmatrix} = C$$

$$c_{ij} \triangleq a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{is}b_{sj}$$

$$\underset{m\times s}{A} \cdot \underset{s\times n}{B} = \underset{m\times n}{C}$$

Matrix - Multiplication

$$c_{i,j} = \sum_{k=1}^{s} a[i][k] * b[k][j]$$

```
inline Matrix operator * (Matrix B) {
    Matrix res;
    res.init(m, B.n);
    for (int i = 1; i <= m; ++i)
        for (int j = 1; j <= B.m; ++j)
            for (int k = 1; k <= n; ++k)
                res.a[i][j] += a[i][k] * B.a[k][j];
    return res;
}
```

Q_1 ：一行乘以一列得到的结果是什么样子？一列乘一行呢？

Q_1 ： 一行乘以一列得到的结果是什么样子？一列乘一行呢？

A_1 ： 一行 × 一列 = 一个数，一列 × 一行 = 一张表。

$$\begin{pmatrix} 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = 14 \qquad \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{pmatrix}$$

Q_2 ： 计算：

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 1 & 3 & 5 \end{pmatrix} \times \begin{pmatrix} 1 & 3 & 3 \\ 2 & 4 & 5 \\ 1 & 5 & 2 \end{pmatrix}$$

Matrix - Multiplication - Quiz

Q_1 ：一行乘以一列得到的结果是什么样子？一列乘一行呢?

A_1 ：一行 $\times$ 一列 $=$ 一个数，一列 $\times$ 一行 $=$ 一张表。

$$\begin{pmatrix} 1 & 2 & 3 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = 14 \qquad \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{pmatrix}$$

Q_2 ：计算:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 1 & 3 & 5 \end{pmatrix} \times \begin{pmatrix} 1 & 3 & 3 \\ 2 & 4 & 5 \\ 1 & 5 & 2 \end{pmatrix} = \begin{pmatrix} 8 & 26 & 19 \\ 12 & 38 & 29 \\ 12 & 40 & 28 \end{pmatrix}$$

矩阵乘法没有交换律

$$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

矩阵乘法有结合律

$$\mathrm{ABC} = (\mathrm{AB})\mathrm{C} = \mathrm{A}(\mathrm{BC})$$

Matrix - Multiplication - Special Conditions

例 $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$

$AB = 0$未必有$A = 0$或$B = 0$

$\begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ -2 & -2 \end{pmatrix}$

AB未必等于BA

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

$AB = AC$未必有$B = C$

定义 若$AB = BA$，则称A和B可交换

定理 设A为$m\times n$矩阵，B、C的维数使下列各式的乘积有定义，则

$$(AB)C = A(BC)$$

$$(A+B)C = AC+BC$$

$$A(B+C) = AB+AC$$

$$(\lambda A)B = \lambda(AB) = A(\lambda B)$$

$$A_{m\times n}I_n = A, \quad I_m A_{m\times n} = A$$

$$\lambda I_n = \begin{pmatrix} \lambda & 0 & \cdots & 0 \\ 0 & \lambda & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda \end{pmatrix}$$

数(纯)量矩阵

注记 $A_{m\times n}(\lambda I_n) = \lambda A,\ (\lambda I_m)A_{m\times n} = \lambda A$

$$A_n(\lambda I_n) = \lambda A = (\lambda I_n)A_n,$$

纯量阵跟任何同阶方阵可交换

矩阵的幂: 矩阵 A 的 k 次幂 A^k 即为 k 个 A 相乘的积。

$$A^1 = A, A^2 = AA, A^k = A^{k-1}A = \underbrace{AA\cdots A}_{k}$$

$$A^0 = I$$

注 若 A 可进行幂运算，则 A 必为方阵。

性质 $A^kA^t = A^{k+t}$, $(A^k)^t = A^{kt}$，注意 $(AB)^k \neq A^kB^k$

计算数的快速幂:

$$a^k = \begin{cases} (a^2)^{\lfloor \frac{k}{2} \rfloor} & \text{if } \ k \bmod 2 = 0 \\ (a^2)^{\lfloor \frac{k}{2} \rfloor} * a & \text{if } \ k \bmod 2 = 1 \end{cases}$$

矩阵乘法满足**结合律**，因此同样可以进行矩阵快速幂加速:

$$A^k = \begin{cases} (A^2)^{\lfloor \frac{k}{2} \rfloor} & \text{if } \ k \bmod 2 = 0 \\ (A^2)^{\lfloor \frac{k}{2} \rfloor} * a & \text{if } \ k \bmod 2 = 1 \end{cases}$$

$$\begin{aligned} \text{e.g.} \quad A^9 &= AAAAAAAAA = (AA)(AA)(AA)(AA)A \\ &= (A^2)(A^2)(A^2)(A^2)A = [(A^2)(A^2)][(A^2)(A^2)]A \\ &= [(A^2)^2]^2A \end{aligned}$$

求解矩阵的幂和 $\mathrm{sum}(n) = \sum_{i=1}^{n} A^i$ 。

· 若 n 为偶数，则（单层复杂度 $O(m^3 \log n)$）:

$$\mathrm{sum}(n) = I + (I + A^{\frac{n}{2}})(A + A^2 + \cdots + A^{\frac{n}{2}})$$

$$= I + (I + A^{\frac{n}{2}})(A(\mathrm{sum}(\frac{n}{2} - 1))$$

· 若 n 为奇数，则（单层复杂度 $O(m^3)$）:

$$\mathrm{sum}(n) = I + A(\mathrm{sum}(n - 1))$$

递归即可，复杂度 $O(m^3 \log^2 n)$ 。

Matrix - Power Sum

- 若 n 为偶数，则（单层复杂度 $O(m^3 \log n)$）：

$$\begin{aligned} sum(n) &= I + (I + A^{\frac{n}{2}})(A + A^2 + \cdots + A^{\frac{n}{2}}) \\ &= I + (I + A^{\frac{n}{2}})(A(sum(\frac{n}{2} - 1)) \end{aligned}$$

- 若 n 为奇数，则（单层复杂度 $O(m^3)$）：

$$sum(n) = I + A(sum(n - 1))$$

```cpp
inline Matrix powsum(Matrix A, int k) {
    Matrix I;
    I.init(A.m, A.m);
    I.id();
    if (k & 1) return I + A * (powsum(A, k - 1));
    return I + (I + A.fpow(k / 2)) * (A * powsum(A, k / 2 - 1));
}
```

转置: 矩阵 A 的转置 A^T 为 A 关于主对角线方向对称所得矩阵。

注 若 A 是 $m \times n$ 阶矩阵，A^T 是 $n \times m$ 阶矩阵。

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

Matrix - Transposition

```cpp
inline Matrix trans() {
    Matrix res;
    //行数列数互换
    res.init(n, m);
    for (int i = 1; i <= m; ++i)
        for (int j = 1; j <= n; ++j)
            res.a[i][j] = a[j][i];
    return res;
}
```

Matrix - Transposition

定义 $$\begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}^T \triangleq \begin{pmatrix} a & d \\ b & e \\ c & f \end{pmatrix}$$

性质

$$(A^T)^T = A$$

$$(A+B)^T = A^T + B^T$$

$$(\lambda A)^T = \lambda A^T$$

$$(AB)^T = B^T A^T$$

Recursive Sequence

Fibonacci

$\mathrm{fib}_0 = \mathrm{fib}_1 = 1$

$\mathrm{fib}_\mathrm{i} = \mathrm{fib}_{\mathrm{i}-1} + \mathrm{fib}_{\mathrm{i}-2}\,(\mathrm{i} \geq 2)$

求第 $\mathrm{k}\,(1 \leq \mathrm{k} \leq 10^{18})$ 项的值（$\bmod\ 10^9 + 7$)?

Fibonacci

$\mathrm{fib}_0 = \mathrm{fib}_1 = 1$

$\mathrm{fib}_\mathrm{i} = \mathrm{fib}_{\mathrm{i}-1} + \mathrm{fib}_{\mathrm{i}-2}\,(\mathrm{i} \geq 2)$

求第 $\mathrm{k}\,(1 \leq \mathrm{k} \leq 10^{18})$ 项的值（$\bmod\ 10^9 + 7$)?

考虑递推向量 $\alpha_\mathrm{i} = [\mathrm{fib}_\mathrm{i}, \mathrm{fib}_{\mathrm{i}-1}]^\mathrm{T} \to \alpha_{\mathrm{i}+1} = [\mathrm{fib}_{\mathrm{i}+1}, \mathrm{fib}_\mathrm{i}]^\mathrm{T}$

转移矩阵 A 满足 $\alpha_{\mathrm{i}+1} = \mathrm{A}\alpha_\mathrm{i}$ 。

$\mathrm{fib}_0 = \mathrm{fib}_1 = 1$

$\mathrm{fib}_i = \mathrm{fib}_{i-1} + \mathrm{fib}_{i-2}\,(i \geq 2)$

求第 $k\,(1 \leq k \leq 10^{18})$ 项的值（$\bmod 10^9+7$）?

考虑递推向量 $\alpha_i = [\mathrm{fib}_i, \mathrm{fib}_{i-1}]^T \to \alpha_{i+1} = [\mathrm{fib}_{i+1}, \mathrm{fib}_i]^T$

转移矩阵 A 满足 $\alpha_{i+1} = A\alpha_i$ 。

$$\begin{bmatrix} \mathrm{fib}_{i+1} \\ \mathrm{fib}_i \end{bmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{bmatrix} \mathrm{fib}_i \\ \mathrm{fib}_{i-1} \end{bmatrix}$$

验证：$\mathrm{fib}_{i+1} = \mathrm{fib}_i + \mathrm{fib}_{i-1}$, $\mathrm{fib}_i = \mathrm{fib}_i$

Recursive Sequence

对于一个递推式，如何确定递推向量？如何确定转移矩阵？

“递归增加法”：找到递推需要的项，加入递推向量。

转移矩阵各行即为每一项对应的递推公式。

e.g. $a_i = 5a_{i-1} - 3a_{i-2} + 7a_{i-4}$

对于一个递推式，如何确定递推向量？如何确定转移矩阵？

“递归增加法”：找到递推需要的项，加入递推向量。

转移矩阵各行即为每一项对应的递推公式。

e.g. $a_i = 5a_{i-1} - 3a_{i-2} + 7a_{i-4}$

$$\begin{bmatrix} a_{i+1} \\ a_i \\ a_{i-1} \\ a_{i-2} \end{bmatrix} = \begin{pmatrix} 5 & -3 & 0 & 7 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{bmatrix} a_i \\ a_{i-1} \\ a_{i-2} \\ a_{i-3} \end{bmatrix}$$

Recursive Sequence - Constant Term

带有常数项的递推如何进行?

e.g. $a_i = 5a_{i-1} - 3a_{i-2} + 7$

Recursive Sequence - Constant Term

带有常数项的递推如何进行?

e.g. $a_i = 5a_{i-1} - 3a_{i-2} + 7$

$$\begin{bmatrix} a_{i+1} \\ a_i \\ 7 \end{bmatrix} = \begin{pmatrix} 5 & -3 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} a_i \\ a_{i-1} \\ 7 \end{bmatrix}$$

$$\begin{bmatrix} a_{i+1} \\ a_i \\ 1 \end{bmatrix} = \begin{pmatrix} 5 & -3 & 7 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} a_i \\ a_{i-1} \\ 1 \end{bmatrix}$$

Recursive Sequence - Constant Power

带有常数的幂次项的递推如何进行?

e.g. $a_i = 5a_{i-1} - 3a_{i-2} + 6^i$

Recursive Sequence - Constant Power

带有常数的幂次项的递推如何进行?

e.g. $a_i = 5a_{i-1} - 3a_{i-2} + 6^i$

$a_{i+1} = 5a_i - 3a_{i-1} + 6^{i+1}$

$$\begin{bmatrix} a_{i+1} \\ a_i \\ 6^{i+1} \end{bmatrix} = \begin{pmatrix} 5 & -3 & 6 \\ 1 & 0 & 0 \\ 0 & 0 & 6 \end{pmatrix} \begin{bmatrix} a_i \\ a_{i-1} \\ 6^i \end{bmatrix}$$

$$\begin{bmatrix} a_{i+1} \\ a_i \\ 6^{i+2} \end{bmatrix} = \begin{pmatrix} 5 & -3 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 6 \end{pmatrix} \begin{bmatrix} a_i \\ a_{i-1} \\ 6^{i+1} \end{bmatrix}$$

带有一次项的递推如何进行?

e.g. $a_i = 5a_{i-1} - 3a_{i-2} + i$

带有一次项的递推如何进行?

e.g. $a_i = 5a_{i-1} - 3a_{i-2} + i$

$a_{i+1} = 5a_i - 3a_{i-1} + i + 1$

$$\begin{bmatrix} a_{i+1} \\ a_i \\ i+1 \\ 1 \end{bmatrix} = \begin{pmatrix} 5 & -3 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} a_i \\ a_{i-1} \\ i \\ 1 \end{bmatrix}$$

Recursive Sequence - Power

带有二次项的递推如何进行?

e.g. $a_i = 5a_{i-1} - 3a_{i-2} + i^2$

带有二次项的递推如何进行?

e.g. $a_i = 5a_{i-1} - 3a_{i-2} + i^2$

$a_{i+1} = 5a_i - 3a_{i-1} + (i+1)^2$

$(i+1)^2 = i^2 + 2i + 1$

$$\begin{bmatrix} a_{i+1} \\ a_i \\ (i+1)^2 \\ i+1 \\ 1 \end{bmatrix} = \begin{pmatrix} 5 & -3 & 1 & 2 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} a_i \\ a_{i-1} \\ i^2 \\ i \\ 1 \end{bmatrix}$$

带有三次、四次项的递推如何进行?

解 $(i+1)^3$, $(i+1)^4$ 展开形式即可。

一道很没有营养的题 [Luogu 1707] 刷题比赛

[NOI 2013] 矩阵游戏

递推一个 $n \times m$ 的矩阵。

$$f_{1,1} = 1$$

$$f_{i,j} = a * f_{i,j-1} + b (j \neq 1)$$

$$f_{i,1} = c * f_{i-1,m} + d (i \neq 1)$$

求 $f[n][m] \bmod 10^9 + 7$ 的值。

数据范围 $n, m \leq 10^{1000000}$, $a, b, c, d \leq 10^9$

把矩阵拼成数列。

转移 A： $f_i = a * f_{i-1} + b$

转移 B： $f_i = c * f_{i-1} + d$

则进行 $m-1$ 次 A 转移，进行 1 次 B 转移，一共 n 轮

[NOI 2013] 矩阵游戏

把矩阵拼成数列。

转移 A： $f_i = a * f_{i-1} + b$

转移 B： $f_i = c * f_{i-1} + d$

递推 $[f_i, 1]^T$，则两个转移矩阵：

$$A = \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} c & d \\ 0 & 1 \end{pmatrix}$$

则所求为 $(A^{m-1}B)^{n-1}A^{m-1}$ 。

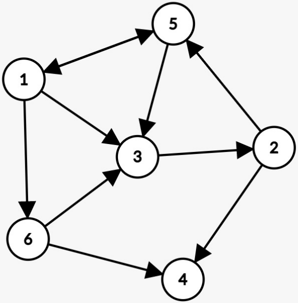
高精度 + 矩阵快速幂 $O(8(\log_n + \log_m))$。

with Graph Theory

Directed Graph

点：图的基本元素

边：点之间的关系，由起点和终点表示（u → v）



1 3
1 5
1 6
2 4
2 5
3 2
5 1
5 3
6 3
6 4

Questions

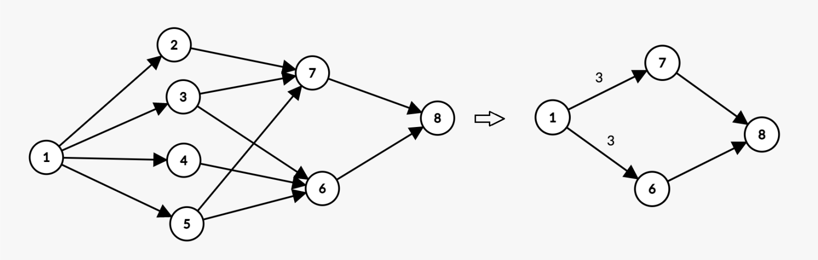
路径：从起点到达终点，所经过的一组边

回路：起点和终点相同的路径

路径的长度（步长）：路径包含的边数

Q1: 从点 a 到点 b 长度为 s 的路径数?

Q2: 从点 a 出发的长度为 s 的回路数?



$b_{(i,\ j)}$ 表示 i 号点指向 j 号点的边的数量。

$a^{s}_{(i,\ j)}$ 表示从 i 号点到 j 号点，长度为 s 的路径数。

$$a^{s}_{(i,\ j)} = \sum_{k=1}^{n} a^{s-1}_{(i,\ k)} \times b_{(k,\ j)}$$

Deduction

$$\begin{cases} a^{s}_{(i,\,1)} & = \sum_{k=1}^{n} a^{s-1}_{(i,\,k)} \times b_{(k,\,1)} \\ a^{s}_{(i,\,2)} & = \sum_{k=1}^{n} a^{s-1}_{(i,\,k)} \times b_{(k,\,2)} \\ & \vdots \\ a^{s}_{(i,\,n)} & = \sum_{k=1}^{n} a^{s-1}_{(i,\,k)} \times b_{(k,\,n)} \end{cases}$$

$$\begin{pmatrix} a^{s}_{(i,\,1)} \\ a^{s}_{(i,\,2)} \\ \vdots \\ a^{s}_{(i,\,n)} \end{pmatrix}^{T} = \begin{pmatrix} a^{s-1}_{(i,\,1)} \\ a^{s-1}_{(i,\,2)} \\ \vdots \\ a^{s-1}_{(i,\,n)} \end{pmatrix}^{T} \begin{pmatrix} b_{(1,\,1)} & b_{(1,\,2)} & \cdots & b_{(1,\,n)} \\ b_{(2,\,1)} & b_{(2,\,2)} & \cdots & b_{(2,\,n)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{(n,\,1)} & b_{(n,\,2)} & \cdots & b_{(n,\,n)} \end{pmatrix}$$

邻接矩阵 B

记 $\overrightarrow{\alpha_i^s} = [\mathsf{a}^s_{(i,\,1)}, \mathsf{a}^s_{(i,\,2)}, \cdots, \mathsf{a}^s_{(i,\,n)}]^T$

则上面的方程组可表示为 $\overrightarrow{\alpha_i^s} = B\,\overrightarrow{\alpha_i^s}$

可见转移矩阵（邻接矩阵）B 与 i 无关。

因此可将 $\overrightarrow{\alpha_1^s}, \overrightarrow{\alpha_2^s}, \cdots, \overrightarrow{\alpha_n^s}$ 的转移同时进行。

记 $A^s = [\,\overrightarrow{\alpha_1^s}, \overrightarrow{\alpha_2^s}, \cdots, \overrightarrow{\alpha_n^s}\,]$，则有 $A^s = BA^{s-1}$

我们称 A^s 是 **步长为 s 的状态矩阵**，则 $A^0 = I_n, A^s = B^s$。

A^s 包含了哪些信息?

Extended Question

Q1: 从点 a 到点 b 长度不超过 s 的路径数?

Q2: 从点 a 出发的长度不超过 s 的回路数?

我们需要查询的矩阵实际上是 $\sum_{i=0}^{s} A^{i}$

这样做的复杂度是 $O(n^3 \times s)$ 的。

幂和可以折半优化到 $O(n^3 \log s)$

[HNOI 2002] 公交车路线

有向图：任意相邻两点之间建双向边，E 没有出边。

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

板子题。

若无停留、爆炸，答案为 $\sum_{1\leq j\leq n} B^t[1][j]$

[TJOI 2017] 可乐

若无停留、爆炸，答案为 $\sum_{1\le j\le n} B^t[1][j]$

停留：每个点连一个自环

爆炸：新建一个爆炸节点 T

每个点都向 T 连边，T 的出边只有自己

[SCOI 2009] 迷路

拆点

假设 $u \to v$ 有一条长度为 w 的边

那么新建 $w-1$ 个虚拟节点用来转移

点数 $10*10*8+10$，$O(n^3 \log t)$ 过不了

如何合并信息相同的节点?

拆点

假设 $u \to v$ 有一条长度为 w 的边

那么新建 $w-1$ 个虚拟节点用来转移

点数 $10*10*8+10$， $O(n^3 \log t)$ 过不了

如何合并信息相同的节点?

把每个点拆成九个点，$u_0, \cdots, u_8$

u_i 表示还有 i 秒可以从 u 离开

连边 $u_8 \to u_7 \to \cdots \to u_0$

$u \to v$ 有一条长度为 w 的边：连边 $u_0 \to v_{w-1}$

点数 $10*9$， $O(n^3 \log t)$ 可过

[ZJOI 2005] 沼泽鳄鱼

对于第 $t-1 \to t$ 秒的转移矩阵 $T^t_{(i,j)}$

· 如果第 t 秒 j 号点有食人鱼，$T^t_{i,j} = 0$

· 否则 $T^t_{i,j} = B_{i,j}$

食人鱼的循环周期为 $\mathrm{lcm}(2,3,4) = 12$ ，转移矩阵 12 秒一循环。

$$\text{ans} = T^{k\%12} \cdots T^0 (T^{11} \cdots T^3 T^2 T^1 T^0)^{\lfloor \frac{k}{12} \rfloor}$$

“不能走刚离开的边”较难处理

考虑拆边后，对边约束，变成“不能走自己的反边”

Trick ：快速找反边的技巧：开始 $\text{tot} = 1$

$$(2k + 1)\ \text{xor}\ 1 = 2k$$

$$(2k)\ \text{xor}\ 1 = 2k + 1$$

点边互换： 求第 $i + 0.5s$ 在某条边上的方案数

建新图：对于两条边 e_1, e_2 都与 u 相连（$e_1 \neq e_2$ ）：

e_1 到 u 的入边 $\rightarrow e_2$ 从 u 的出边

e_2 到 u 的入边 $\rightarrow e_1$ 从 u 的出边

设 S_1 表示所有 A 的出边集合

S_2 表示所有 B 的入边集合

$$ans = \forall e_1 \in S_1, e_2 \in S_2, \sum A^{t-1}[e_1][e_2]$$

Gauss-Jordan Elimination

已知 n 元一次线性方程组：

$$\begin{cases} a_{1,1}x_1 & + & a_{1,2}x_2 & + \cdots + & a_{1,n}x_n & = b_1 \\ a_{2,1}x_1 & + & a_{2,2}x_2 & + \cdots + & a_{2,n}x_n & = b_2 \\ \vdots & & \vdots & \ddots & \vdots & \vdots \\ a_{m,1}x_1 & + & a_{m,2}x_2 & + \cdots + & a_{m,n}x_n & = b_m \end{cases}$$

- 若方程组无解，输出"No Solution"
- 若方程组唯一解，输出该唯一解
- 若方程组有无穷多解，输出通解的参数向量形式 *

Gauss-Jordan Elimination

高斯-约当消元法，算法竞赛中常用的解线性方程组的算法。

1. 枚举主元列为第 i 列
2. 枚举目前未被其他主元占用的行，找到主元系数不为 0 的任意一行，将该行换到第 i 行
3. 将其余所有行对应位置消成 0

时间复杂度 $O(n^3)$： 枚举主元 $O(n)$ ，消元 $O(n^2)$

思考一下算法实现过程中的问题。

Cont'd

Q_1：何时方程组无解？何时方程组有无穷多解？

Cont'd

Q_1： 何时方程组无解？何时方程组有无穷多解？

A_1： 无解：存在某一行系数全部为 0，但常数列不为 0。

多解：方程组有解，且存在一行系数和常数列全部为 0。

Cont’d

Q_1： 何时方程组无解？何时方程组有无穷多解？

A_1： 无解：存在某一行系数全部为 0，但常数列不为 0。

多解：方程组有解，且存在一行系数和常数列全部为 0。

Q_2： 无穷多解时，如何求通解的参数向量形式？

Cont'd

Q_1： 何时方程组无解？何时方程组有无穷多解？

A_1： 无解：存在某一行系数全部为 0，但常数列不为 0。

多解：方程组有解，且存在一行系数和常数列全部为 0。

Q_2： 无穷多解时，如何求通解的参数向量形式？

A_2： 求出齐次线性方程组的通解，加上非齐次线性方程组的一个特解（可以是最开始的常数列）

Cont'd

Q_1： 何时方程组无解？何时方程组有无穷多解？

A_1： 无解：存在某一行系数全部为 0，但常数列不为 0。

多解：方程组有解，且存在一行系数和常数列全部为 0。

Q_2： 无穷多解时，如何求通解的参数向量形式？

A_2： 求出齐次线性方程组的通解，加上非齐次线性方程组的一个特解（可以是最开始的常数列）

Q_3： 为避免精度误差，选取主元行应当按照什么样的原则？

Cont'd

Q_1： 何时方程组无解？何时方程组有无穷多解？

A_1： 无解：存在某一行系数全部为 0，但常数列不为 0。

多解：方程组有解，且存在一行系数和常数列全部为 0。

Q_2： 无穷多解时，如何求通解的参数向量形式？

A_2： 求出齐次线性方程组的通解，加上非齐次线性方程组的一个特解（可以是最开始的常数列）

Q_3： 为避免精度误差，选取主元行应当按照什么样的原则？

A_3： 选取主元列对应系数最大的一行

[SDOI 2006] 线性方程组

已知 n 元一次线性方程组：

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n = b_2 \\ \vdots \qquad \vdots \qquad \ddots \qquad \vdots \qquad \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n = b_n \end{cases}$$

- 若方程组无解，输出"-1"
- 若方程组唯一解，输出该唯一解
- 若方程组有无穷多解，输出"0"

模版题，实现细节见笔记代码。

n 维欧几里得球的概念：

球心 $(x_1, x_2, \cdots, x_n)$，半径 r

球上的点 $(a_1, a_2, \cdots, a_n)$ 满足 $\sum_{i=1}^{n}(a_i - x_i)^2 = r^2$

从 $n+1$ 个点中任选一个点记作特殊点 $A(A_1, A_2, \cdots, A_n)$

剩下的 n 个点坐标记作 $a_i(a_{i,1}, a_{i,2}, \cdots, a_{i,n})$

$$\sum_{k=1}^{n}(a_{i,k}^2 + x_k^2 - 2a_{i,k}x_k) = r^2 (1)$$

$$\sum_{k=1}^{n}(A_k^2 + x_k^2 - 2A_kx_k) = r^2 (2)$$

从 $n+1$ 个点中任选一个点记作特殊点 $A(A_1, A_2, \cdots, A_n)$

剩下的 n 个点坐标记作 $a_i(a_{i,1}, a_{i,2}, \cdots, a_{i,n})$

$$\sum_{k=1}^{n}(a_{i,k}^2 + x_k^2 - 2a_{i,k}x_k) = r^2 (1)$$

$$\sum_{k=1}^{n}(A_k^2 + x_k^2 - 2A_kx_k) = r^2 (2)$$

对 $\forall i \in [1, n]$，令对应的 $(1) - (2)$：

$$\sum_{k=1}^{n}(a_{i,k}^2 - A_k^2 - 2(a_{i,k} - A_k)x_k) = 0$$

解线性方程组即可。

[ICPC 2020 Jinan Regional] A

给定两个 n 阶 01 方阵 A, B，求有多少个 n 阶 01 方阵 C，满足:

$$A * C \mod 2 = B \& C$$

数据范围 $1 \leq n \leq 200$

各列贡献独立，分别计数，答案相乘。

只有 0 和 1 在膜 2 意义下运算：

· $\mathrm{a and b} = \mathrm{a} * \mathrm{b}$

· $\mathrm{a\,xor\,b} = \mathrm{a} \pm \mathrm{b} \pmod 2$

约束条件可转化为线性方程组（见板书）

假设自由元个数为 x，该列方案数即为 2^{x}

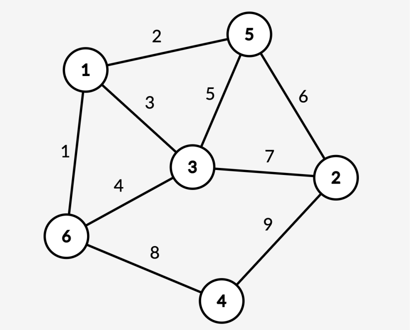
异或高斯消元可用 bitset 优化，总复杂度 $\mathrm{O}(\mathrm{n}^4/64)$

Matrix-Tree

Incidence Matrix

对于 n 个点 m 条边的无向图，我们定义关联矩阵 G：

对于图中第 k 条无向边 (a, b)，令 $G_{(a,k)} = -1, G_{(b,k)} = 1$



Incidence Matrix

对于 n 个点 m 条边的无向图，我们定义关联矩阵 G：

对于图中第 k 条无向边 (a, b)，令 $G_{(a,k)} = -1, G_{(b,k)} = 1$

$$G = \begin{pmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 \\ 0 & 0 & 1 & -1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

（上页图对应关联矩阵）

定义基尔霍夫矩阵 $K = GG^T$ 。

$$GG^T_{(i,j)} = \sum_{k=1}^{n} G_{(i,k)} G^T_{(k,j)} = \sum_{k=1}^{n} G_{(i,k)} G_{(j,k)}$$

即 G 第 i 行和第 j 行的点积。

如果 $i = j$，则 $GG^T_{(i,i)} = deg_i$

如果 $i \neq j$，若存在边 (i, j)，则 $GG^T_{(i,j)} = -1$，否则 $= 0$

定义度数矩阵 D ，$D_{(i,i)} = deg_i$，其他位置为 0

可以发现基尔霍夫矩阵的一个性质 $K = D - B$

（无向图）矩阵树定理：

对于一个 n 个点的无向图，求基尔霍夫矩阵。

$\forall i \in [1, n]$，将基尔霍夫矩阵删掉第 i 行第 i 列。

求得到的新矩阵的行列式的值，即为对应原图的生成树个数。

模版题：[HEOI 2015] 小 Z 的房间

* 有向图的矩阵树定理：树形图计数（内向树形图、外向树形图）

* 变元矩阵树定理：[SDOI 2014] 重建

Questions?

Thanks for listening.

Blog	blog.gyx.me
Email	sgcolin@163.com

Use $\LaTeX$