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）

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

Matrix - Definition

当行数和列数相同时，即 $\mathrm{m} = \mathrm{n}$ 时，称其为 n **阶方阵**。

记号：主对角线（$\mathrm{a}_{1,1} \to \mathrm{a}_{\mathrm{n,n}}$），副对角线（$\mathrm{a}_{1,\mathrm{n}} \to \mathrm{a}_{\mathrm{n},1}$）

$$\begin{pmatrix} \mathrm{a}_{1,\,1} & \mathrm{a}_{1,\,2} & \cdots & \mathrm{a}_{1,\,\mathrm{n}} \\ \mathrm{a}_{2,\,1} & \mathrm{a}_{2,\,2} & \cdots & \mathrm{a}_{2,\,\mathrm{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{a}_{\mathrm{n},\,1} & \mathrm{a}_{\mathrm{n},\,2} & \cdots & \mathrm{a}_{\mathrm{n},\,\mathrm{n}} \end{pmatrix}$$

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));
    }

};
```

Matrix - Special Matrixs

· 零矩阵 $O_{m\times n}$： 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 $\Lambda_n = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$

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

Matrix - Addition & Subtraction

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）。

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

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

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;
}
```

Matrix - Scalar Multiplication

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

```
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;
}
```

Matrix - Summary (1)

定理 设A, B, C是同型的矩阵，λ，μ为数，则

$$A + B = B + A$$

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

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

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

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

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

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

Matrix - Multiplication

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 的行数 = B 的列数。

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

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

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

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;
}
```

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？

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 × 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 × 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：

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

Matrix - Multiplication

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

Matrix - Summary (2)

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

Matrix - Power

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 $(AB)^k \neq A^kB^k$

Matrix - Fast Power

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：

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

e.g. $A^9 = AAAAAAAAA = (AA)(AA)(AA)(AA)A$

$= (A^2)(A^2)(A^2)(A^2)A = [(A^2)(A^2)][(A^2)(A^2)]A$

$= [(A^2)^2]^2 A$

Matrix - Fast Power

```
inline void id() {
    for (int i = 1; i <= m; ++i) a[i][i] = 1;
}

inline Matrix fpow(int t) {
    Matrix res;
    res.init(m, m);
    res.id();
    Matrix tmp = *this;
    for(; t; t >>= 1, tmp = tmp * tmp)
        if (t & 1) res = res * tmp;
    return res;
}
```

Matrix - Transposition

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。

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

Matrix - Transposition

```
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

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

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

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

Fibonacci

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

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

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

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 $\alpha_{i+1} = A\alpha_i$ 。

Fibonacci

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

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

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

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

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

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

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

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

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

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

Recursive Sequence - Power

带有一次项的递推如何进行?

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

Recursive Sequence - Power

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

Recursive Sequence - Power

带有二次项的递推如何进行?

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

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

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

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

Recursive Sequence - Power

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

"Matrix"

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

”Matrix”

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

"Matrix"

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 $O(8(\log_n + \log_m))$。

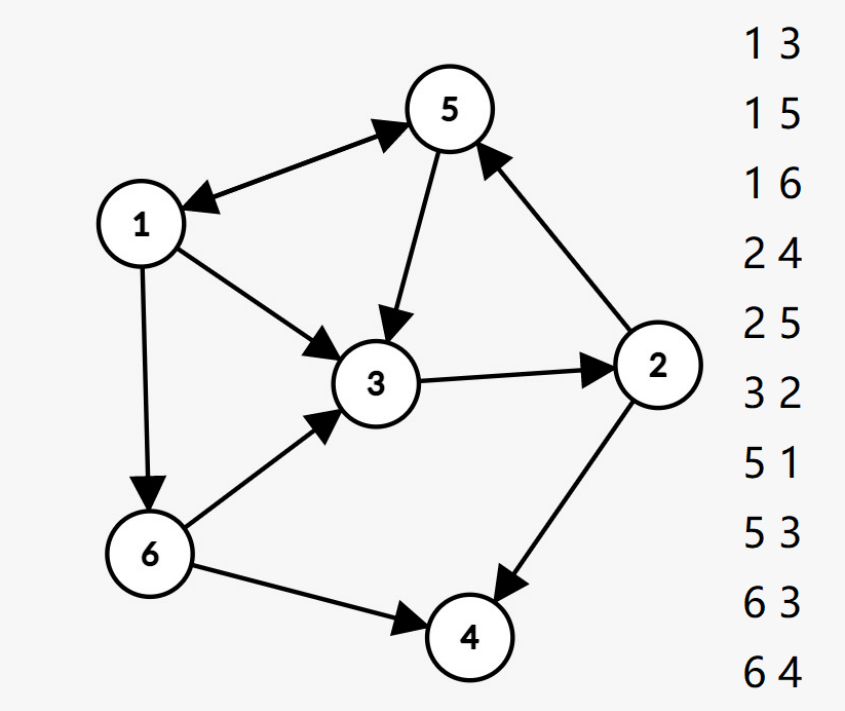
原题 [NOI 2013] 矩阵游戏

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（$u \to v$）



Questions

路径：从起点到达终点，所经过的一组边

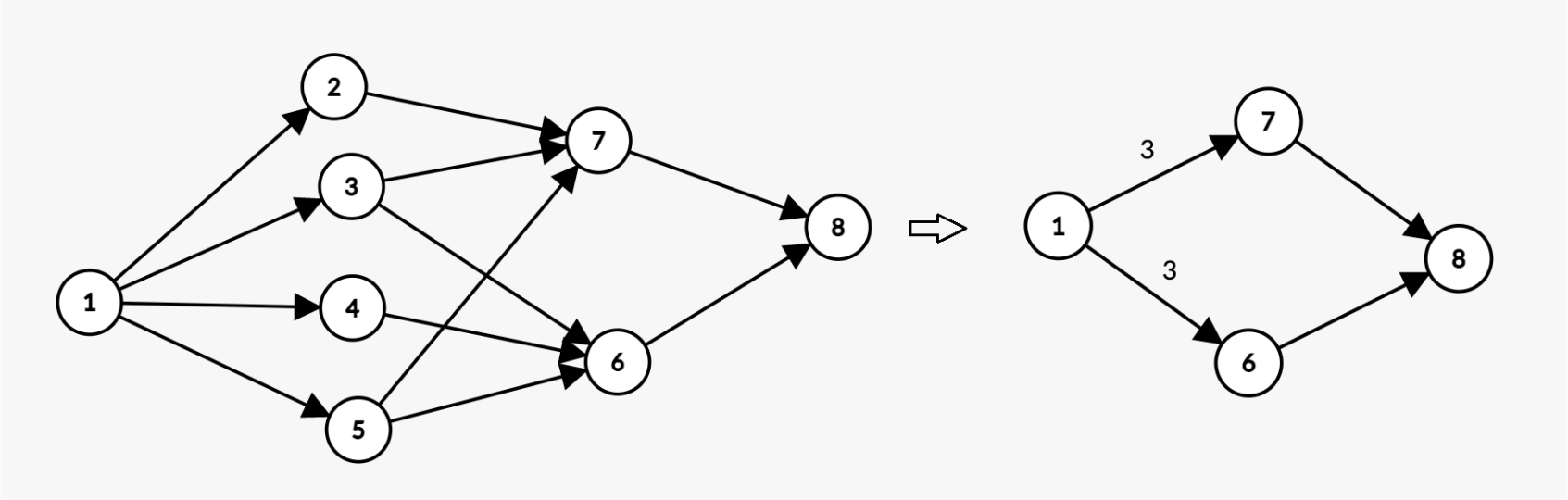
回路：起点和终点相同的路径

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?

Counting Tricks



$b_{(i,j)}$ 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

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

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

邻接矩阵 B

Deduction - Cont'd

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

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。

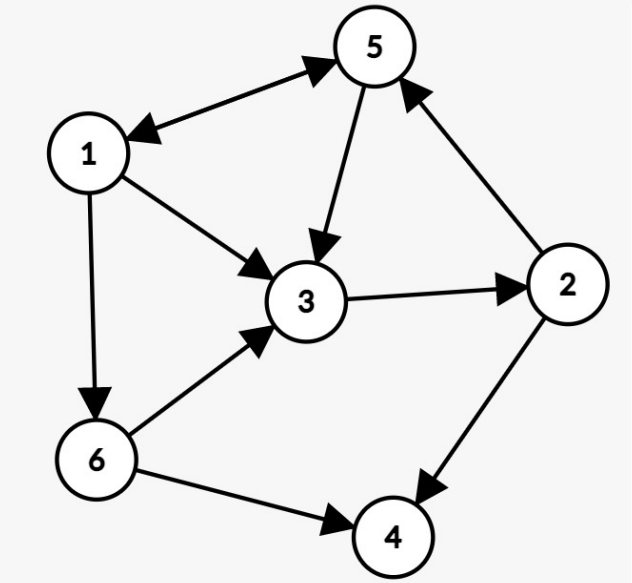
因此可将 $\overrightarrow{\alpha_1^s}, \overrightarrow{\alpha_2^s}, \cdots, \overrightarrow{\alpha_n^s}$ 的转移同时进行。

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？

Example



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

图　　邻接矩阵

Example - Cont'd

$$A^0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad A^1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

$$A^2 = \begin{pmatrix} 1 & 1 & 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad A^3 = \begin{pmatrix} 0 & 2 & 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

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。

[TJOI 2017] 可乐

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

[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?

[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?

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

Questions?

Summary

Thanks for listening.

Blog blog.gyx.me

Email sgcolin@163.com

Use LaTeX