



# 矩阵

用于快速进行递推，常与快速幂结合使用。

矩阵作为一种存储向量的方式，是一种必须的数据结构。

## Code

注意（重载输入输出流时）：要记得返回 `istream` 或 `ostream`，并且仔细思考，哪些需要引用传参，哪些不能使用引用传参。

```

struct Matrix {
    int size_x, size_y, res[maxn][maxn];
    inline void clear() {memset(res, 0, sizeof(res)); }
    Matrix creatM(int size_x, int size_y){
        Matrix ep;
        ep.size_x = size_x;
        ep.size_y = size_y;
        ep.clear();
        return ep;
    }
    friend Matrix operator * (Matrix A, Matrix B) {
        Matrix ans=ans.creatM(A.size_x,B.size_y);
        if(A.size_y != B.size_x)return ans;
        for (int i = 1; i <= A.size_x; i++)
            for (int j = 1; j <= B.size_y; j++)
                for (int k = 1; k <= A.size_y; k++)
                    ans.res[i][j] = (ans.res[i][j] + (A.res[i][k] % mod) * (B.res[k][j] % mod)) % mod;
        return ans;
    }
    friend istream& operator >> (istream& stream, Matrix& A){
        for(int i=1;i<=A.size_x;i++)
            for(int j=1;j<=A.size_y;j++)
                stream>>A.res[i][j];
        return stream;
    }
    friend ostream& operator << (ostream& stream, Matrix A){
        for(int i=1;i<=A.size_x;i++){
            for(int j=1;j<=A.size_y;j++){
                stream<<A.res[i][j]<<' ';
            }
            stream<<'\n';
        }
        return stream;
    }
};

```

## 例题1

P3390 【模板】矩阵快速幂

[Code](#)

## 例题2

P1939 矩阵加速（数列）

通过这道题看懂矩阵快速幂的应用

1. 观察题目给出的式子

$$a_x = \begin{cases} 1 & x \in \{1, 2, 3\} \\ a_{x-1} + a_{x-3} & x \geq 4 \end{cases}$$

2. 设计出关于  $a_x$  的矩阵

$$\begin{bmatrix} a_x \\ a_{x-1} \\ a_{x-2} \end{bmatrix}$$

3. 设计转移矩阵：列表看系数

	$a_{x-1}$	$a_{x-2}$	$a_{x-3}$
$a_x$	1	0	1
$a_{x-1}$	1	0	0
$a_{x-2}$	0	1	0

[Explain](#)

$$a_x = 1 \times a_{x-1} + 0 \times a_{x-2} + 1 \times a_{x-3}$$

$$a_{x-1} = 1 \times a_{x-1} + 0 \times a_{x-2} + 0 \times a_{x-3}$$

$$a_{x-2} = 0 \times a_{x-1} + 1 \times a_{x-2} + 0 \times a_{x-3}$$

把系数填到表里我们就得到了转移矩阵

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

[Code](#)