矩阵幂和

$\sum_{i=0}^{n} A^i$

$I + A + A^2 + \cdots + A^n$

若 n 为偶数：$I + (I + A^{\frac{n}{2}})(A + A^2 + \cdots + A^{\frac{n}{2}})$

若 n 为奇数：$ans = I + A(ans(\text{偶数时的答案}))$

O（$m^3 \log n)$

```
inline matrix sum(matrix A, int k) {
  if (k & 1) return I + A * (sum(A, k - 1));
  return I + (I + A.fpow(k / 2))(A * sum(A, k / 2 - 1));
}
```

步数不超过 s 的路径条数

查询的是 $0, 1, 2, \cdots, s$ 步从 a 到 b 的路径条数

$ans = A_{a,b}^0 + A_{a,b}^1 + A_{a,b}^2 + \cdots + A_{a,b}^s$

$S = \sum_{i=0}^{s} A^s \Rightarrow ans = S_{a,b}$ 求邻接矩阵的幂和

$a_i = 2a_{i-3} + a_{i-4}$

$a_{i+1} = 2a_{i-2} + a_{i-3}$

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

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

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

$$\begin{bmatrix} a_{i+1} \\ a_i \\ 6^{i+1} \end{bmatrix} = \begin{pmatrix} 5 & -3 & 30 \\ 1 & 0 & 0 \\ 0 & 0 & 6 \end{pmatrix} \begin{bmatrix} a_i \\ a_{i-1} \\ 6^i \end{bmatrix}$$

[NOI2013] 矩阵游戏

A 转移：$[f_i, 1]^T$

$$\begin{bmatrix} f_{i+1} \\ 1 \end{bmatrix} = \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} \begin{bmatrix} f_i \\ 1 \end{bmatrix}$$

B 转移：$[f_i, 1]^T$

$$\begin{bmatrix} f_{i+1} \\ 1 \end{bmatrix} = \begin{pmatrix} c & d \\ 0 & 1 \end{pmatrix} \begin{bmatrix} f_i \\ 1 \end{bmatrix}$$

合并两种转移?

[SCOI 2009] 迷路

```
int n = rd();
int m = rd();
int totid = 0;
int id[N][9];

for (int i = 1; i <= n; ++i)
  for (int j = 0; j <= 8; ++j) id[i][j] = ++totid;

for (int i = 1; i <= m; ++i) {
  int u = rd();
  int v = rd();
  int w = rd();
  add(id[u][0], id[v][w - 1]);
}

//....

int S = rd();
int T = rd();
ans = A.a[id[S][0]][id[T][0]];
```

[SDOI 2009] HH去散步

```
for (int i = 0; i <= tot; ++i) {
  int u = e[i].to;
  for (int j = hd[u]; j; j = e[j].nxt)
    if ((i ^ 1) != j) ++a[i][j];
}

S1:
for(int i = hd[S]; i; i = e[i].nxt) s1.add(i);
S2:
for (int i = 0; i <= tot; ++i)
```

```
  if (e[i].to == T) s2.add(i);
for(int i = 1; i <= s1.size(); ++i)
  for (int j = 1; j <= s2.size(); ++j)
    ans += a[s1[i]][s2[j]];
```

[ZJOI 2005] 沼泽鳄鱼

```
matrix B, T[12];
vector<int> s;
int fish[N][4], r[N];

inline int pos(int id, int t) {
  return fish[id][t % r[id]];
}

for (int i = 1; i <= m; ++i) {
  r[i] = rd();
  for (int j = 0; j < r[i]; ++j) fish[i][j] = rd();
}

matrix S = I;

for (int t = 0; t < 12; ++t) {
  s.clear();
  T[t] = B;
  for (int i = 1; i <= m; ++i) s.push_back(pos(i,t));
  for (int i = 0; i < len; ++i) {
    int u = s[i];
    for (int j = 1; j <= n; ++j) T[t].a[j][u] = 0;
  }
  S = S * T[t];
}

S = S.fpow(k / 12);
k = k % 12;
for (int i = 0; i < k; ++i) S = T[i] * S;
```

加减（高斯）消元法

$$
\begin{cases}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
\cdots\cdots\cdots\cdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m
\end{cases}
$$

$\forall i > 1$,第 i 行整体减掉第一行的 $\frac{a_{i1}}{a_{11}}$ 倍

$$\begin{pmatrix} 1000 & a & b & | & c \\ 0.5 & d & e & | & f \end{pmatrix}$$

$$eps = 0.01$$

$$r2 += (0.5/1000)r1$$

$$\delta = \frac{0.5 + eps}{1000 + eps} - \frac{0.5}{1000} = 0.01$$

$$\delta = \frac{1000 + eps}{0.5 + eps} - \frac{1000}{0.5} = 40$$

[SDOI 2006] 线性方程组

模版题

```
#include <cstdio>
#include <cctype>
#include <cmath>
#include <iostream>
#include <cstring>
#include <cstdlib>
#include <algorithm>
#define N 107
using namespace std;

inline int rd(){
  int x = 0;
  bool f = 0;
  char c = getchar();
  for(; !isdigit(c); c = getchar())
    if (c == '-') f = 1;
  for (; isdigit(c); c = getchar())
    x = x * 10 + (c ^ 48);
  return f ? -x : x;
}

double a[N][N];

bool vis[N];

const double eps = 1e-8;

int main() {
  int n = rd();
  for (int i = 1; i <= n; ++i)
    for (int j = 1; j <= n + 1; ++j) a[i][j] = rd();
```

```
  double tmpmax;
  int maxr;
  for (int i = 1; i <= n; ++i) {
    //找第i列系数最大的行
    tmpmax = 0; maxr = i;
    for (int j = 1; j <= n; ++j)
      if (!vis[j] && abs(a[j][i]) > tmpmax) {
        tmpmax = abs(a[j][i]); maxr = j;
      }

    //如果这一列系数全部为0跳过
    if (tmpmax <= eps) continue;

    //将系数最大的行换到第i行
    if (maxr != i) swap(a[i], a[maxr]);
    vis[i] = 1;

    //对其他行关于第i列消元
    for (int j = 1; j <= n; ++j)
      if (i != j) {
        double t = a[j][i] / a[i][i];
        for (int k = i; k <= n + 1; ++k)
          a[j][k] -= t * a[i][k];
      }
  }

  bool f = 0;
  for (int i = 1; i <= n; ++i)
    if (abs(a[i][i]) <= eps) {
      f = 1;
      if (abs(a[i][n + 1]) > eps) {
        puts("-1"); return 0;
      }
    }

  if (f) {puts("0"); return 0;}
  for (int i = 1; i <= n; ++i)
    printf("x%d=%.2lf\n", i, a[i][n + 1] / a[i][i]);

  return 0;
}
```

[JSOI 2008] 球形空间产生器

n 维球体：$(x_1,x_2,\cdots,x_n),r$

n 个点：第 i 个点的坐标 $(a_{i,1},a_{i,2},\cdots,a_{i,n})$

第 $n+1$ 个点 $A(A_1,A_2,\cdots,A_n)$

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

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

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

$$\forall i\in[1,n],(1)-(2):$$

$$\sum_{k=1}^{n}(a_{i,k}^2-A_k^2-2(a_{i,k}-A_k)x_k)=0(3)$$

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：

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

数据范围 $n\le 200$

矩阵乘法：对于 D 的一列，C 中产生影响的 只有对应一列

与操作：对于 D 的一列，C 中产生影响的 只有对应一列

各列的贡献是独立的

对各列方案计数，答案就是各列答案的积。

$C=[c_1,c_2,\cdots,c_n]$

$B=[b_1,b_2,\cdots,b_n]$

$A*c_i \mod 2=b_i\&c_i$

$$A * c_i = \begin{pmatrix} a_{1,\ 1} & a_{1,\ 2} & \cdots & a_{1,\ n} \\ a_{2,\ 1} & a_{2,\ 2} & \cdots & a_{2,\ n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,\ 1} & a_{n,\ 2} & \cdots & a_{n,\ n} \end{pmatrix} \begin{pmatrix} c_{1,i} \\ c_{2,i} \\ \vdots \\ c_{n,i} \end{pmatrix}$$

$$b_i \& c_i = \begin{pmatrix} b_{1,i} \& c_{1,i} \\ b_{2,i} \& c_{2,i} \\ \vdots \\ b_{n,i} \& c_{n,i} \end{pmatrix}$$

对于所有的行，第 j 行：$\sum_{k=1}^{n} a_{j,k} c_{k,i} - b_{j,i} * c_{j,i} = 0 \pmod 2$

联立得 n 个未知量 $c_{1,i}, \cdots, c_{n,i}$ ， n 个方程的线性方程组

自由变量个数是 x ， 合法的解就有 2^x 种。

自由变量的个数形如 00000000 的行数

只有 0 和 1 在膜 2 意义下运算：

- $a \& b = a * b$
- $a \text{ xor } b = a \pm b \pmod 2$$
- 复杂度 $\mathrm{O}(n^4/64) = 1e8/16$

```
bitset<N> A[N],B[N],M[N];

inline int count(int col) {

  for (int i = 1; i <= n; ++i) {
    for (int j = 0; j <= n; ++j) M[i][j] = A[i][j];
    M[i][i] = M[i][i] ^ B[i][col];
  }

  for (int i = 1; i <= n; ++i) {
    for (int k = i; k <= n; ++k)
      if (M[k][i]) {
        if (k != i) swap(M[k], M[i]);
        break;
      }
    if (!M[i][i]) continue;
    for (int k = 1; k <= n; ++k)
      if (k != i && M[k][i]) M[k] ^= M[i];
  }

  int ans = 0;
  for (int i = 1; i <= n; ++i) ans += (M[i].count() == 0);
  return ans;
```

```
}
```

行列式

定义在方阵上的。

把这个方阵消元，得到只有对角线上有数

$$A = \begin{pmatrix} a_{1,\ 1} & 0 & \cdots & 0 \\ 0 & a_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n,\ n} \end{pmatrix}$$

定义 $|A| = \prod_{i=1}^{n} a_{i,i}$

基尔霍夫矩阵

$$K = GG^T_{i,j} = \sum_{k=1}^{n} G_{i,k} G^T_{k,j} = \sum_{k=1}^{n} G_{i,k} G_{j,k}$$
$$K = D - B$$

（无向图）矩阵树定理

对于一个 n 个点的无向图，生成树的个数

$\forall i \in [1, n]$ 基尔霍夫矩阵删掉第一行第一列，得到的新矩阵的行列式的值

[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

拟阵-线性基

推荐学习链接 https://oi.men.ci/linear-basis-notes/