E!5!-G题-重建猫娘乐园

21371258 冯睿冰

元素均在{0,1,2}的 3×3 的矩阵，可以通过以下三种操作，调整元素的排列顺序，直至其变成指定的组合。

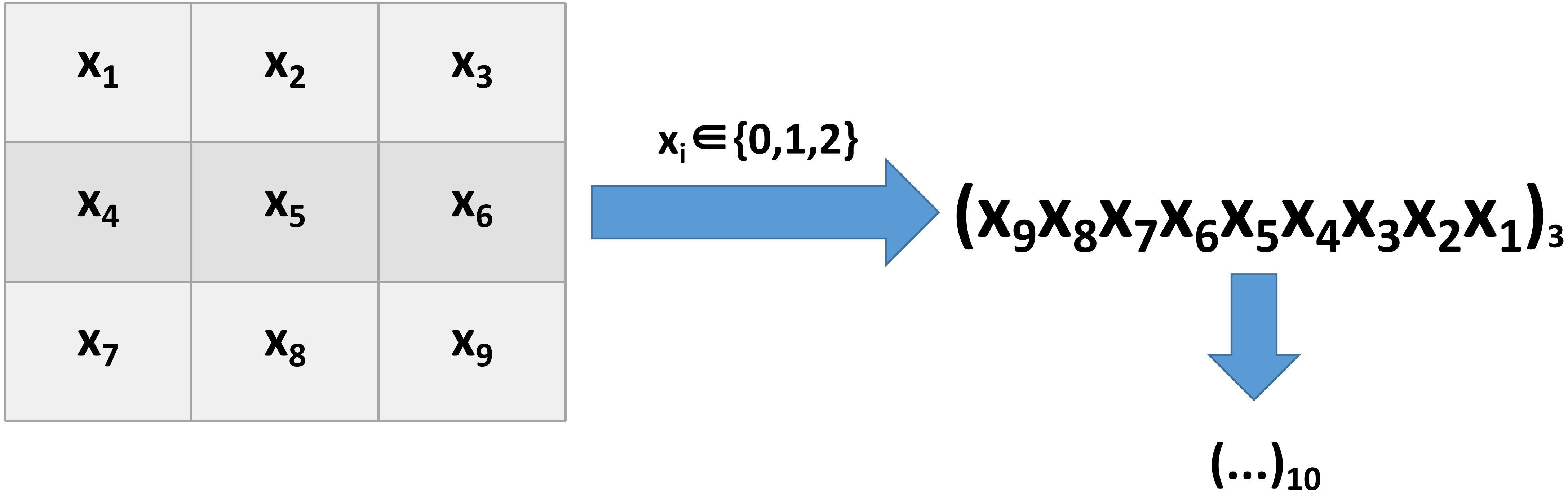
1. 「旋转」：花费 a 枚摩拉，选择任意一个 2×2 的子矩阵，将其顺时针旋转 90。

2. 「对换」：花费 b 枚摩拉，选择相邻的两个元素，并交换它们的位置。

3. 「预制」：花费 c 枚摩拉，选择一个 n×m 的子矩阵，将它预制成给定的样式。

求需要花费的最少的摩拉数

将每个矩阵转换为一个唯一对应的三进制数
共有3^9=19683个



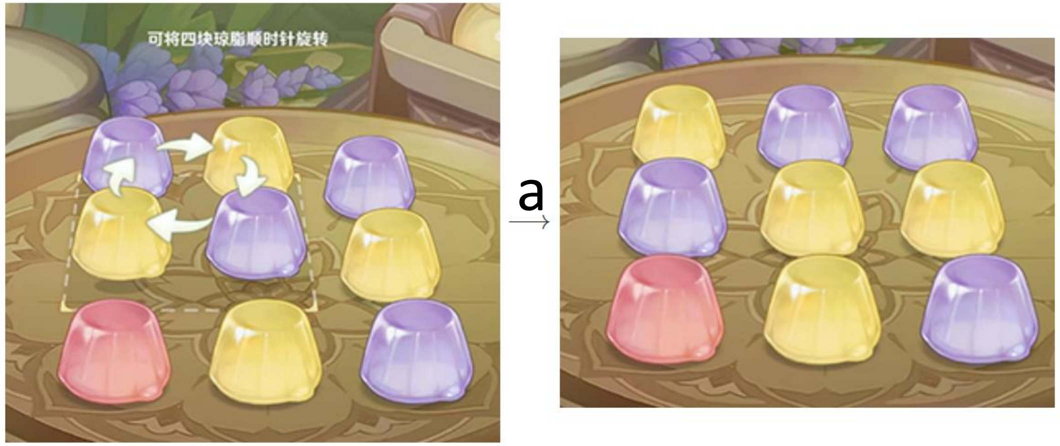
x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

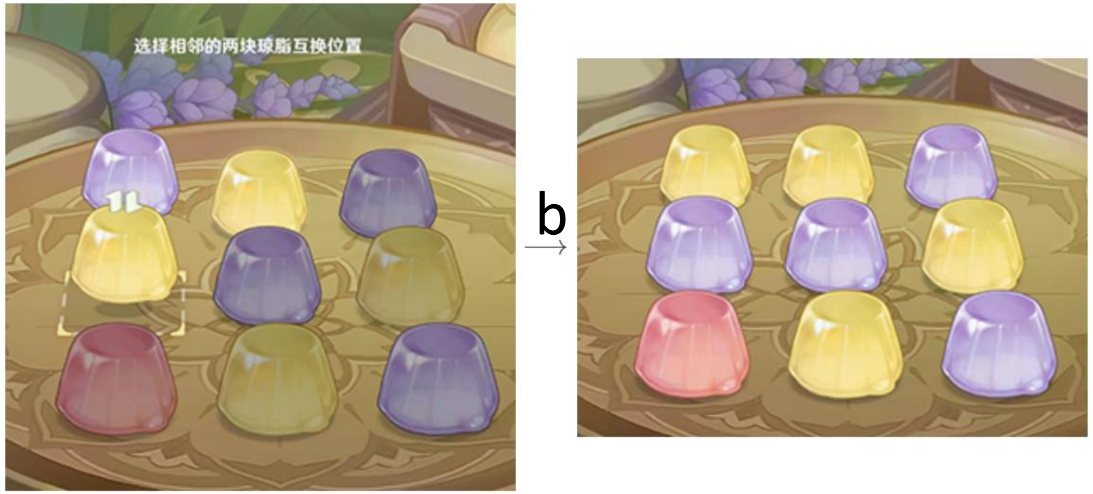
$x_i \in \{0,1,2\}$

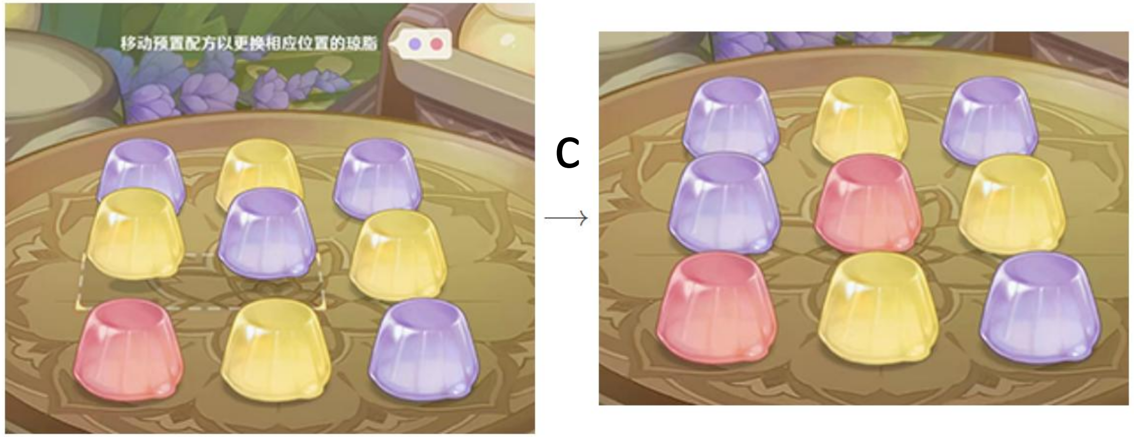
$(x_9x_8x_7x_6x_5x_4x_3x_2x_1)_3$

$(\ldots)_{10}$

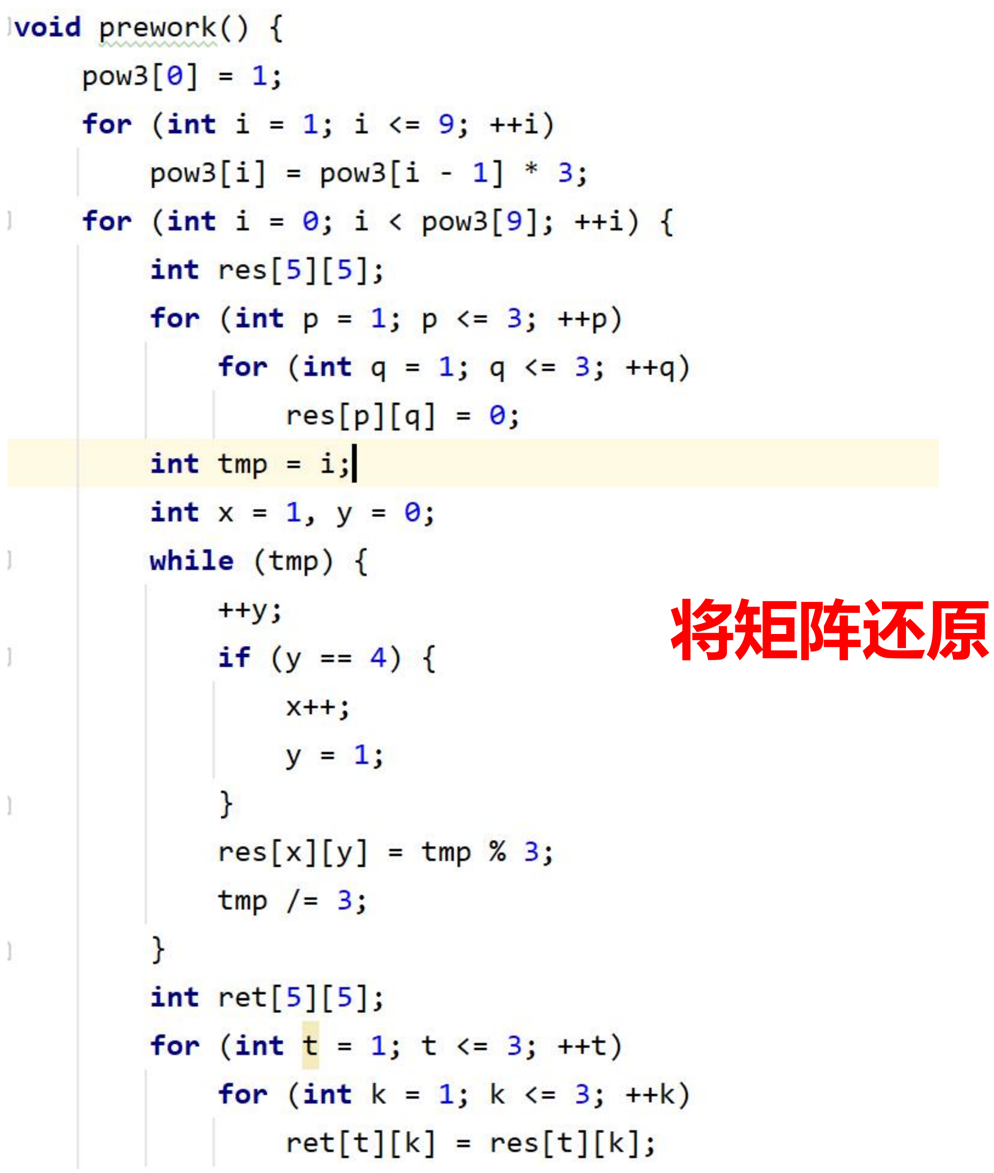
再将每个矩阵通过对应操作得到的矩阵
用一条权值为a/b/c的有向边连接，最终形成一个有向图





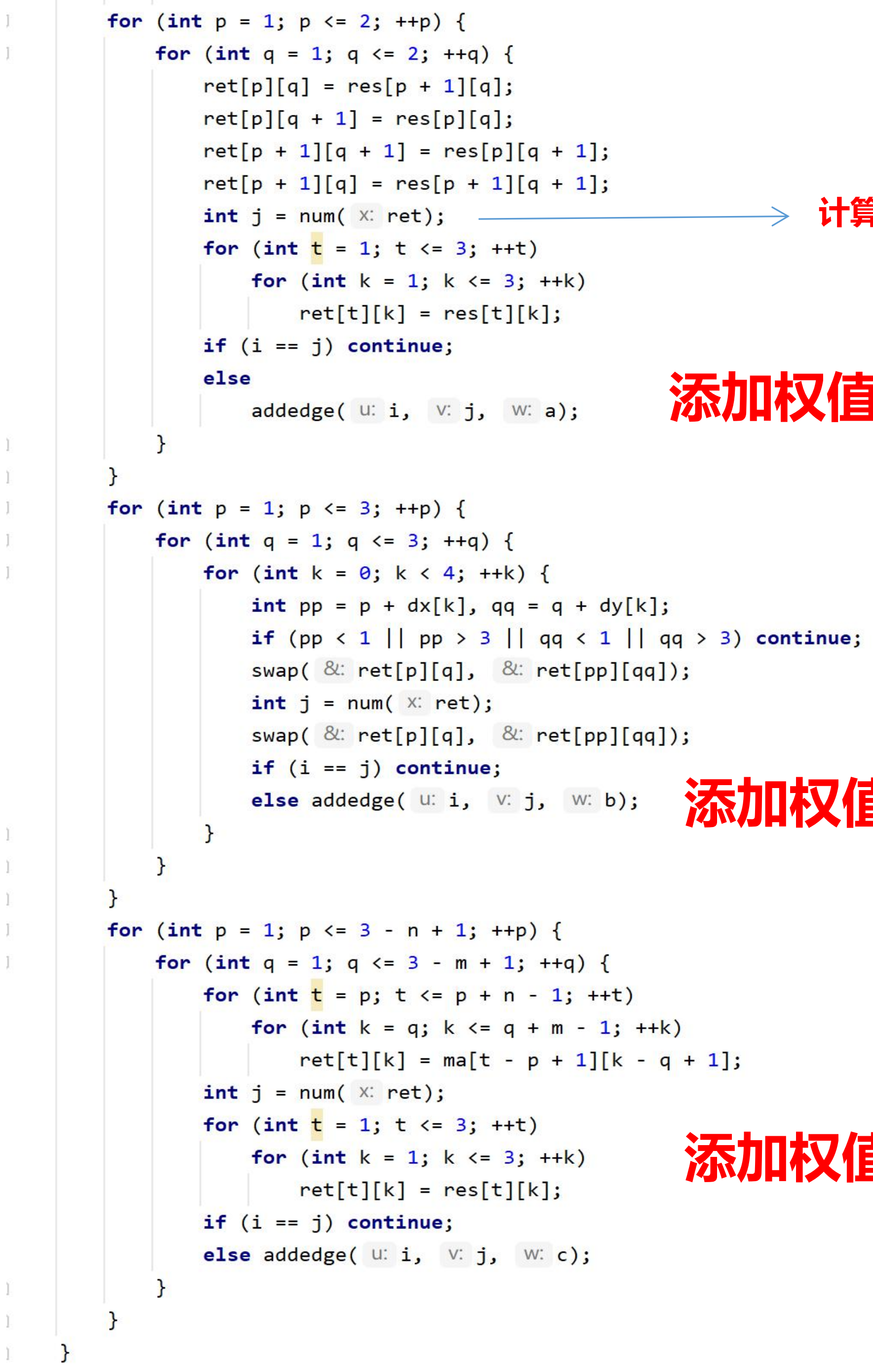


原问题转化为求两点间的最短路！



```
void prework() {
    pow3[0] = 1;
    for (int i = 1; i <= 9; ++i)
        pow3[i] = pow3[i - 1] * 3;
    for (int i = 0; i < pow3[9]; ++i) {
        int res[5][5];
        for (int p = 1; p <= 3; ++p)
            for (int q = 1; q <= 3; ++q)
                res[p][q] = 0;
        int tmp = i;
        int x = 1, y = 0;
        while (tmp) {
            ++y;
            if (y == 4) {
                x++;
                y = 1;
            }
            res[x][y] = tmp % 3;
            tmp /= 3;
        }
        int ret[5][5];
        for (int t = 1; t <= 3; ++t)
            for (int k = 1; k <= 3; ++k)
                ret[t][k] = res[t][k];
```

将矩阵还原



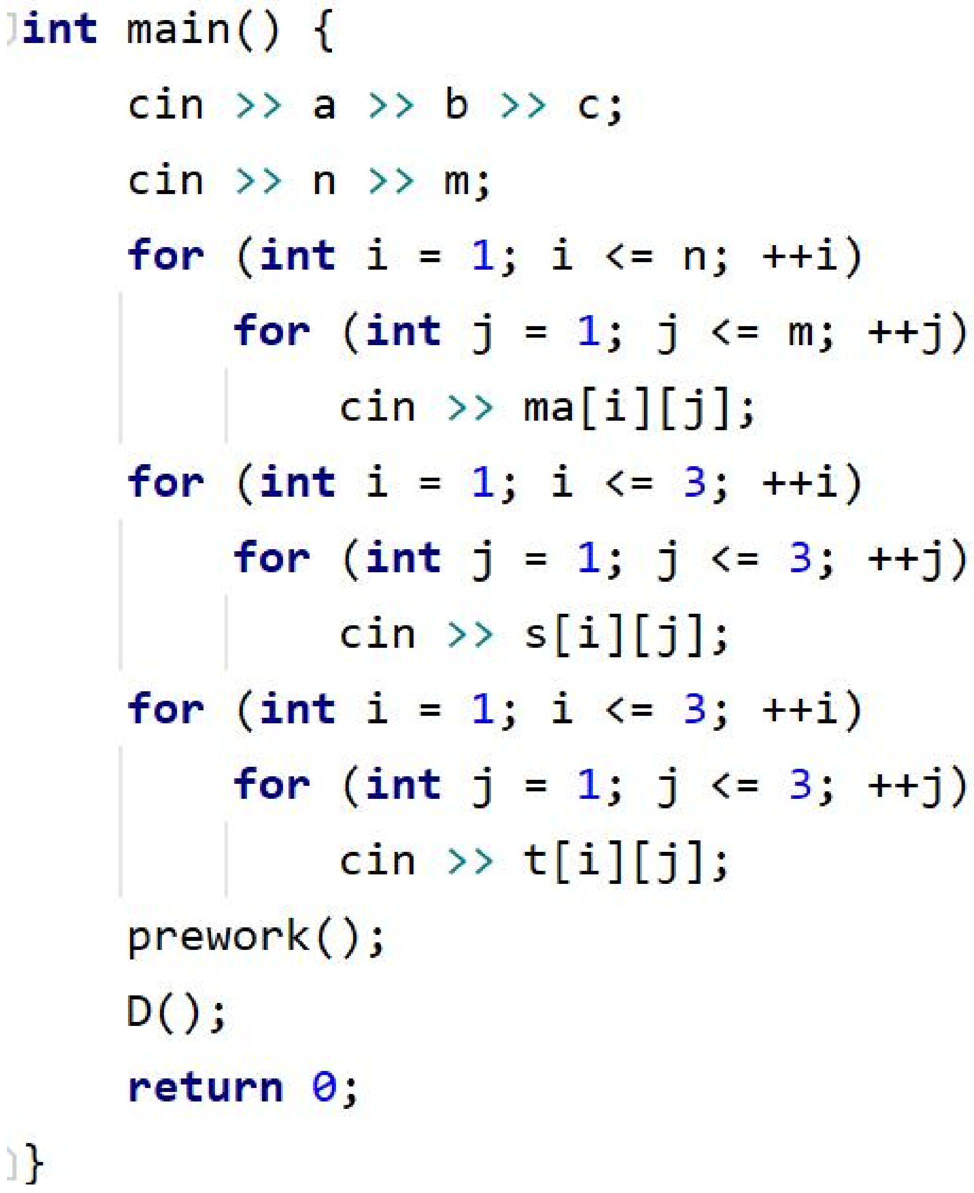
```
        for (int p = 1; p <= 2; ++p) {
            for (int q = 1; q <= 2; ++q) {
                ret[p][q] = res[p + 1][q];
                ret[p][q + 1] = res[p][q];
                ret[p + 1][q + 1] = res[p][q + 1];
                ret[p + 1][q] = res[p + 1][q + 1];
                int j = num( x: ret);
                for (int t = 1; t <= 3; ++t)
                    for (int k = 1; k <= 3; ++k)
                        ret[t][k] = res[t][k];
                if (i == j) continue;
                else
                    addedge( u: i,  v: j,  w: a);
            }
        }
        for (int p = 1; p <= 3; ++p) {
            for (int q = 1; q <= 3; ++q) {
                for (int k = 0; k < 4; ++k) {
                    int pp = p + dx[k], qq = q + dy[k];
                    if (pp < 1 || pp > 3 || qq < 1 || qq > 3) continue;
                    swap( &: ret[p][q],  &: ret[pp][qq]);
                    int j = num( x: ret);
                    swap( &: ret[p][q],  &: ret[pp][qq]);
                    if (i == j) continue;
                    else addedge( u: i,  v: j,  w: b);
                }
            }
        }
        for (int p = 1; p <= 3 - n + 1; ++p) {
            for (int q = 1; q <= 3 - m + 1; ++q) {
                for (int t = p; t <= p + n - 1; ++t)
                    for (int k = q; k <= q + m - 1; ++k)
                        ret[t][k] = ma[t - p + 1][k - q + 1];
                int j = num( x: ret);
                for (int t = 1; t <= 3; ++t)
                    for (int k = 1; k <= 3; ++k)
                        ret[t][k] = res[t][k];
                if (i == j) continue;
                else addedge( u: i,  v: j,  w: c);
            }
        }
    }
```

计算所得矩阵对应的数

添加权值为a的有向边

添加权值为b的有向边

添加权值为c的有向边



```cpp
int main() {
    cin >> a >> b >> c;
    cin >> n >> m;
    for (int i = 1; i <= n; ++i)
        for (int j = 1; j <= m; ++j)
            cin >> ma[i][j];
    for (int i = 1; i <= 3; ++i)
        for (int j = 1; j <= 3; ++j)
            cin >> s[i][j];
    for (int i = 1; i <= 3; ++i)
        for (int j = 1; j <= 3; ++j)
            cin >> t[i][j];
    prework();
    D();
    return 0;
}
```