高消第二讲

高消 基本知识提高

$$Ax = b, x = A^{-1}b$$

其实高斯消元的另外一个作用就是就 A 的逆矩阵，如果 A 有逆的情况下，对

$$AI ---\text{高消}----\Rightarrow IA^{-1}$$

然后可以通过矩阵的逆来直接计算

The Pilots Brothers' refrigerator

Time Limit: 1000MS Memory Limit: 65536K
Total Submissions: 18651 Accepted: 7149 Special Judge

Description

The game “The Pilots Brothers: following the stripy elephant” has a quest where a player needs to open a refrigerator.

There are 16 handles on the refrigerator door. Every handle can be in one of two states: open or closed. The refrigerator is open only when all handles are open. The handles are represented as a matrix 4x4. You can change the state of a handle in any location **[i, j]** ($1 \le i, j \le 4$). However, this also changes states of all handles in row **i** and all handles in column **j**.

The task is to determine the minimum number of handle switching necessary to open the refrigerator.

Input

The input contains four lines. Each of the four lines contains four characters describing the initial state of appropriate handles. A symbol “+” means that

the handle is in closed state, whereas the symbol “–” means “open”. At least one of the handles is initially closed.

Output

The first line of the input contains N – the minimum number of switching. The rest N lines describe switching sequence. Each of the lines contains a row number and a column number of the matrix separated by one or more spaces. If there are several solutions, you may give any one of them.

Sample Input

```
-+--
----
----
-+--
```

Sample Output

```
6
1 1
1 3
1 4
4 1
4 3
4 4
```

Source

Northeastern Europe 2004, Western Subregion

必须一题多解，用求逆的方法打表直接过。

```
//poj 2965 使用逆元打表，直接算   陈宇
#include <iostream>
#include <cstring>
#include <cstdio>
//#define yu
using namespace std;
```

```
int a[16][33],c[16][16];  //系数矩阵
int ans[16];
void gauss()//消元
{
    for(int i=0;i<16;i++)//i 代表列，也是主元的位置
     {
        int k=i;    //k 代表行，从对角线的行开始就行
        for(;k<16;k++)
            if(a[k][i]!=0) //找到这列第 1 个不为 0 的行，好做主元啊
                break;
        for(int j=0;j<=31;j++) //交换整行
            swap(a[i][j],a[k][j]);
       //开始消元
        for(int j=0;j<16;j++)   //j 代表行
            if(i!=j&&a[j][i])  //不是主元行，要消的行已经是 1 才消
                for(int k=0;k<=31;k++)  //k 代表列
                    a[j][k]=a[i][k]^a[j][k];
    }
    //下面把逆元放到数组 c 中
     for(int i=0;i<16;i++)
         for(int j=0;j<16;j++)
        c[i][j]=a[i][j+16];


}
 int init()
 {
```

```
  memset(a,0,sizeof(a));
 int k=0;
   for(int i=0;i<4;i++)
      for(int j=0;j<4;j++)
      {
         k=i*4+j;

            for(int p=0;p<4;p++)
            {
               a[i*4+p][k]=1;
               a[p*4+j][k]=1;
            }

      }
   //下面要再A 的右边的加上一个单位阵


   for(int i=0;i<16;i++)
       a[i][16+i]=1;

  return 0;

 }

int main()
{
   char c1[20];
    init();
    int pt[16];
    memset(pt,0,sizeof(pt));
   for(int i=0;i<4;i++)
     {

        cin>>c1;
       for(int j=0;j<4;j++)
       {
          if (c1[j]=='+') pt[i*4+j]=1;

       }

     }
```

```
    gauss();
#ifdef yu
    for(int i=0;i<16;i++){
    for(int j=0;j<=31;j++)
    printf("%d ",a[i][j]);
    printf("\n");
    }
#endif

//下面直接矩阵相乘  x=A^(-1) * b;

for(int i=0;i<16;i++)
    for(int j=0;j<16;j++)
    ans[i]=ans[i]^(c[i][j]*pt[j]);


    int sum=0;
   for(int i=0;i<16;i++ ) sum+=ans[i];
   cout<<sum<<endl;
   for (int i=0;i<16;i++)
   {
       if (ans[i]==1)
        {
           int x=i/4+1;
           int y=i%4+1;
         cout<<x<<" "<<y<<endl;
        }
   }







    return 0;
}
```

---　　打表直接 AC

```
#include <iostream>
```

```cpp
#include <cstring>
//#define yu
using namespace std;

int main()
{
int c[16][16]={1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
0, 0, 0,
1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0,
1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,
1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0,
0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0,
0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0,
0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0,
0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,
0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0,
0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1,
0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1,
0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1,
0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1,};
   char c1[20];
   // init();
    int pt[16],ans[16];

    memset(pt,0,sizeof(pt));
    memset(ans,0,sizeof(ans));
   for(int i=0;i<4;i++)
     {

        cin>>c1;
       for(int j=0;j<4;j++)
       {
          if (c1[j]=='+') pt[i*4+j]=1;

       }

     }
      // gauss();
    #ifdef yu
       for(int i=0;i<16;i++){
       for(int j=0;j<=31;j++)
```

```
        printf("%d ",a[i][j]);
        printf("\n");
        }
    #endif

    //下面直接矩阵相乘  x=A^(-1) * b;

    for(int i=0;i<16;i++)
        for(int j=0;j<16;j++)
        ans[i]=ans[i]^(c[i][j]*pt[j]);


        int sum=0;
       for(int i=0;i<16;i++ ) sum+=ans[i];
       cout<<sum<<endl;
       for (int i=0;i<16;i++)
       {
           if (ans[i]==1)
            {
               int x=i/4+1;
               int y=i%4+1;
             cout<<x<<" "<<y<<endl;
            }
       }
    return 0;
}
```

练习 1　把 poj 1222 用逆矩阵的方法，A 掉；

练习 2　用本 GAUSS（）　自己编写 二进制枚举自由变量的题目

练习 3　Nefu　503　504　505　506 题

练习 4　杭电　4952　Boring Game 给 7 天时间

#