鳥取大学教育学部
研究報告（自然科学），
42（1993），31-43

最適配置のプログラム

後　藤　和　雄・大　平　正　雄

Kazuo GOTO and Masao OHIRA: A computer program of optimum arrangement

（1993年4月20日　受理）

1　は　じ　め　に

オペレーションズリサーチ（米語では，Operation Research, OR, 英語では，Operational Research という）は，第2次世界大戦中に，イギリスで対空火器の研究に関して生まれ，すぐにアメリカにもその思想が伝わり，米国で活発に研究された。軍事行動の効果を最大にするための研究に，多くの数学者たちがかかわり，物資の効率的輸送，生産性の向上，最適配置といった問題を解決することに端を発したものである。戦後，これらのノウハウを持った人たちが，企業の利益を上げるという目的に利用できることに気がつき，大いに研究されて，今日に至っている。ORは，その後，一般化され，ある体系があり，その運営に関して起こるいろいろな問題を科学的，数学的に解決するための具体的な研究として位置づけられている。

日本には，戦争中には，このような科学的，数学的戦略がなかったので，いろいろと無駄や失敗もあったようである。しかし，戦後になり工業が立ち直るにつれ，技術者や研究者が，米国のオペレーションズリサーチの考え方を導入し，現在ではいろいろな場所で使用されている。

ORの数学的モデルには，配分，競争，待ち合わせ，在庫，生産計画モデルなどがある。ORによる解の求めるアルゴリズムは，

(1)　問題の定式化
(2)　モデルの作成と解の探索
(3)　モデルと解との比較検討
(4)　解を管理し実施すること

からなる。

線型計画法は，いろいろな本でよく扱われており，また，コンピュータプログラムも書籍に載っている。この線型計画法の歴史は，少なくとも G. Monge (1781) や J. B. Fourier (1823) まで遡ることができるようである。

この論文では，最適配置の問題をとり上げ，それを効率的に解決するハンガリー法を紹介し，そのコンピュータプログラムを Basic と C 言語で作成したのでここに載せて，使用する人の便宜をはかることを目的としている。Basic は，*N_{88}-DiskBasic*, C 言語は Turbo C を用いている。このプログラムの特徴は，すべての解の組み合わせが求められることである。テクニカルな所もあるが，モジュールに分解し，分かりやすく書いてある。

2 最適配置の問題

ある n 個のものを n ヶ所に1つずつ配置をするとき，i のものを j の場所に配置するコストが a_{ij} 必要とする。これを行列表示すると次のようになる。

$$\begin{array}{c} \\ i \end{array}\begin{array}{c} \begin{array}{c} j \end{array} \\ \begin{pmatrix} \vdots \\ \cdots\ a_{ij}\ \cdots \\ \vdots \end{pmatrix} \end{array}$$

行列Aで考えても，もとの問題と同値となるから，以後，行列で考える。

n 次正方行列Aの配置とは，Aの成分より n 個を取りだした数の組

$$(a_{1p(1)},\ a_{2p(2)},\ \cdots,\ a_{np(n)})$$

（ただし，$p(1), \cdots, p(n)$ は，$\{1, \cdots, n\}$ の置換）で，どの行からも，どの列からもちょうど1個の成分が入っているように取られたものである。これは，8クイーンの配置の問題と関連している。このような配置の数は，行列Aが n 次正方行列とすると $n!$ とおりの配置が存在する。

1つの配置 $(b_1,\ b_2, \cdots,\ b_n)$ が最適であるとは配置の値

$$b_1 + b_2 + \cdots + b_n$$

が最小となるときをいう。（符号を反転すると，最大値になるときも求めることができる）有限の組み合わせしかないから，最適な配置は常に存在するが，一意に決まるとはかぎらない。また，$n!$ の組み合わせがあるので，これは n が大きくなると急に大きくなりすべての組み合わせを求めて計算するというには，時間がかかり困難に直面する。たとえば，$20! = 2.4329 \times 10^{18} \fallingdotseq$ 約243.29京（1京＝10000兆）で不可能に近い。これを劇的に解決する方法として，ハンガリー法（cf. [1], [3]）がある。そのアルゴリズムを書くと次のようになる。

手続

1. 各行の最小数をみつけて，その行の各成分からその数を引く
2. 各列の最小数をみつけて，その列の各成分からその数を引く
   （1, 2は順序が逆でもよい）
3. 行と列の上にいくつかの線をすべての0の上を通るように引く。
   このとき，線の本数は最も少ないようにする。
4. （最適性）
   (a) 引いた線の数が n ならば0のみからなる配置を1つみつけてこのプログラムを終わる。（この論文では，すべての解を求めるプログラムを作成している）
   (b) 引かれた線が n 本未満ならば5へ
5. 線が引かれていない成分の中で最小数 $M>0$ をみつける。
6. 線が引かれていない成分から M を引き，線の交点の成分には M を加え，1へもどる。

具体例で行ってみると次のようになる。

例1.

$$\begin{pmatrix} 4 & 6 & 9 \\ 7 & 9 & 9 \\ 5 & 3 & 2 \end{pmatrix} \rightarrow \text{手続1} \rightarrow \begin{pmatrix} 0 & 3 & 7 \\ 3 & 6 & 7 \\ 1 & 0 & 0 \end{pmatrix} \rightarrow \text{手続2, 3} \rightarrow \begin{pmatrix} \textcircled{0} & 3 & 7 \\ 0 & 3 & 4 \\ 1 & 0 & 0 \end{pmatrix} \rightarrow \text{手続4 } (b) \rightarrow$$

$$\rightarrow \text{手続5 } (M=3) \rightarrow \text{手続6} \rightarrow \begin{pmatrix} 0 & 0 & 4 \\ 0 & 0 & 1 \\ 4 & 0 & 0 \end{pmatrix} \rightarrow \text{手続1, 2, 3} \rightarrow \begin{pmatrix} \textcircled{0} & 0 & 4 \\ 0 & 0 & 1 \\ 4 & 0 & 0 \end{pmatrix} \rightarrow 4\ (a) \rightarrow$$

$$\rightarrow \begin{pmatrix} \textcircled{0} & 0 & 4 \\ 0 & \textcircled{0} & 1 \\ 4 & 0 & \textcircled{0} \end{pmatrix}, \quad \begin{pmatrix} 0 & \textcircled{0} & 4 \\ \textcircled{0} & 0 & 1 \\ 4 & 0 & \textcircled{0} \end{pmatrix}.$$

よって，求める最適配置は

$$(a_{11},\ a_{22},\ a_{33}) \text{ or } (a_{12},\ a_{21},\ a_{33})$$

である。

以下に Basic 言語と C 言語を用いて開発したプログラムと実行例を掲載する。

References

[1] 伊藤昇・他：行列とその応用，紀伊国屋書店 (1987).

[2] 岩堀長慶：線型不等式とその応用 (岩波講座基礎数学)，岩波書店 (1977).

[3] T. L. Saaty: Mathematical methods of operations rsearch, McGraw-Hill, 1959 (邦訳) オペラーションズリサーチの数学的方法，上，下，紀伊国屋書店，1960.

Abstract

We make a computer program for optimum arrangement in two variable linear programing by using Hungarian method (cf. [3]). We use N_{88}-Disk Basic (NEC) or Turbo C (Borland) to develop the program.

プログラム実行例

```
正方行列の次数を入力してください 3

 1 行 1 列の値は 4
 1 行 2 列の値は 6
 1 行 3 列の値は 9

 2 行 1 列の値は 7
 2 行 2 列の値は 9
 2 行 3 列の値は 9

 3 行 1 列の値は 5
 3 行 2 列の値は 3
 3 行 3 列の値は 2

   1 行     4   6   9
   2 行     7   9   9
   3 行     5   3   2


正しいですか Y／N ?Y
---手続   1---
---手続   2---
---手続   3---
     0     1     5
     0     1     2
     3     0     0
```

```
線の数  2
線の数  2
---手続   4---
---手続   5---
---手続   1---
---手続   2---
---手続   3---
     0     0     4
     0     0     1
     4     0     0
線の数  1
線の数  2
線の数  3
---手続   4---


     4     6     9
     7     9     9
     5     3     2
この配列の最適配置は
(  1   1)(  2   2)(  3   3)
(  1   2)(  2   1)(  3   3)
 2 個  解答があります
```

BASIC言語プログラムリスト

```
1000 '*****************************************************************
1010 '**                                                             **
1020 '**       ハンガリー法で最適の配置を求めるプログラム            **
1030 '**                          すべての答えを求める               **
1040 '**                                                             **
1050 '**                                                             **
1060 '*****************************************************************
1070  DIM A(50,50),B(50,50),C(50,50),P(50),Q(50),J(50),F(50)
1080  DIM AI(50),A1(50),AJ(50),A2(50),P1(50),PD(50),Q1(50),QD(50)
1090  CLS
1100 INPUT "正方行列の次数を入力してください" ; N
1110  FOR I = 1 TO N
1120   FOR J = 1 TO N
1130     PRINT I;"行";J;"列の値は";:INPUT A(I,J)
1140     B(I,J)=A(I,J)
1150   NEXT J
1160     PRINT
1170  NEXT I
1180 '
1190  GOSUB *HYOU ' 配置行列の表示と訂正
1200 '
1210 *TETU '---------手続  1   -----------------
1220       PRINT "---手続   1 --- "
1230 FOR I=1 TO N
1240    MIN=A(I,1)
1250   FOR J=1 TO N
1260    IF MIN > A(I,J) THEN MIN=A(I,J)
1270   NEXT J
1280   FOR J=1 TO N
1290    A(I,J)=A(I,J)-MIN
1300   NEXT J
1310 NEXT I
1320 '---------------手続   2-------------------
```

```
1330        PRINT "---手続   2---"
1340 FOR J=1 TO N
1350     MIN=A(1,J)
1360    FOR I=1 TO N
1370     IF MIN>A(I,J) THEN MIN=A(I,J)
1380    NEXT I
1390    FOR I=1 TO N
1400     A(I,J)=A(I,J)-MIN
1410    NEXT I
1420 NEXT J
1430 '---------------手続   3--------------------
1440       PRINT "---手続   3---"
1450  C=0 :MAX=0 :GOSUB *ZERO
1460  FOR I=1 TO N
1470    P(I)=0:Q(I)=0
1480   FOR J=1 TO N
1490    IF A(I,J)=0 THEN A(I,0)=A(I,0)+1
1500   NEXT J
1510    IF A(I,0)>MAX THEN MAX=A(I,0)
1520  NEXT I
1530  FOR J=1 TO N
1540   FOR I=1 TO N
1550    IF A(I,J)=0 THEN A(0,J)=A(0,J)+1
1560   NEXT I
1570    IF A(0,J)>MAX THEN MAX=A(0,J)
1580  NEXT J
1590       V=1 :GOSUB *HYOUJI
1600 '
1610 FOR M=MAX TO 1 STEP -1
1620 '
1630    GOSUB *DNYUURYOKU
1640    GOSUB *GYOU
1650    GOSUB *RETU
1660   FOR I=1 TO N
1670    A1(I)=AI(I) :A2(I)=AJ(I) :P1(I)=PD(I) :Q1(I)=QD(I)
1680   NEXT I
1690  C1=CC
1700 '
1710    GOSUB *DNYUURYOKU
1720    GOSUB *RETU
1730    GOSUB *GYOU
1740 '
1750  IF C1 >= CC THEN GOSUB *PASE1
1760  IF C1 < CC  THEN GOSUB *PASE2
1770  C1=0
1780       PRINT "線の数 ";C
1790 NEXT M
1800 '-------------------手続   4-------------------
1810           PRINT "---手続   4---"
1820    IF C=N THEN *HAITI
1830 '-------------------手続   5-----------------
1840           PRINT "---手続   5---"
1850  FOR I=1 TO N
1860  FOR J=1 TO N
1870   IF (P(I)<>1) AND (Q(J)<>1) THEN MINS=A(I,J):I=N:J=N
1880  NEXT J
1890  NEXT I
1900 '~~~~~線が引かれていない成分から最小数を決める~~~~
1910  FOR I=1 TO N
1920     IF P(I)=1 THEN *HH
1930   FOR J=1 TO N
1940     IF Q(J)=1 THEN *EE
1950     IF MINS>A(I,J) THEN MINS=A(I,J)
1960    *EE
1970   NEXT J
1980    *HH
1990   NEXT I
2000 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
2010  FOR I=1 TO N   ' 行列のすべての成分から
2020  FOR J=1 TO N   '     Mを引く
2030   A(I,J)=A(I,J)-MINS
2040  NEXT J
2050  NEXT I
2060 '~~~~~~~~~~~~~~6-2~~~~~~~~~~~~~~~~~~~~~~~
2070  FOR I=1 TO N           ' 線を引いた行の成分は
2080    IF P(I)<>1 THEN *FF      ' にはMを加える
2090   FOR J=1 TO N
2100    A(I,J)=A(I,J)+MINS
2110   NEXT J
2120   *FF
2130  NEXT I
2140 '~~~~~~~~~~~~~~6-3~~~~~~~~~~~~~~~~~~~~~~~
2150  FOR J=1 TO N           ' 線を引いた列の成分には
2160    IF Q(J)<>1 THEN *GG     '  Mを加える
2170   FOR I=1 TO N
2180    A(I,J)=A(I,J)+MINS
2190   NEXT I
2200   *GG
2210  NEXT J
2220 '
2230   GOTO *TETU   ' 手続  1  へ
2240 '
2250 *HAITI'----0のみの配置を捜す----------------
2260 PRINT:PRINT:V=0:GOSUB *HYOUJI
2270   PRINT "この配列の最適配置は"
2280    I=1:J(I)=0
2290 *LOOP
2300   F(J(I))=0
2310   IF J(I)=N THEN J(I)=0 :I=I-1 :GOTO *LOOP
2320    FOR Z=J(I)+1 TO N
2330     IF ( F(Z)<>0 )OR( A(I,Z)<>0 ) THEN *SKIP
2340     J(I)=Z :J(I+1)=0 :F(Z)=1 :Z=N :I=I+2
2350       *SKIP
2360    NEXT Z
2370 '
2380  I=I-1
2390    IF I>N THEN GOSUB *ANS
2400    IF I>0 THEN *LOOP
2410   PRINT KEI;"個  解答があります"
2420 END
2430 '
2440 '--------------------サブルーチン--------------------
2450 *HYOUJI
2460  FOR I=1 TO N
2470  FOR J=1 TO N
2480   IF V=0 THEN C(I,J)=B(I,J) ELSE IF V=1 THEN C(I,J)=A(I,J)
2490   PRINT USING " ####";C(I,J);
2500  NEXT J
2510   PRINT
2520  NEXT I
2530 RETURN
2540 '
2550 *ZERO
2560   FOR I=1 TO N
2570    A(I,0)=0 :A(0,I)=0
2580   NEXT I
2590 RETURN
2600 '
2610 *GYOU
2620  FOR I=1 TO N
2630    IF AI(I) <> M THEN *AA ELSE PD(I)=1:CC=CC+1
2640   FOR J=1 TO N
2650    IF A(I,J)=0 THEN AI(I)=AI(I)-1 :AJ(J)=AJ(J)-1
2660   NEXT J
2670    *AA
2680  NEXT I
```

```
2690 RETURN
2700 '
2710 *RETU
2720  FOR J=1 TO N
2730    IF AJ(J) <> M THEN *BB ELSE QD(J)=1:CC=CC+1
2740   FOR I=1 TO N
2750    IF A(I,J)=0 THEN AI(I)=AI(I)-1 :AJ(J)=AJ(J)-1
2760   NEXT I
2770    *BB
2780  NEXT J
2790 RETURN
2800 '
2810 *DNYUURYOKU
2820  FOR I=1 TO N
2830   AI(I)=A(I,0) :AJ(I)=A(0,I) :PD(I)=P(I) :QD(I)=Q(I)
2840  NEXT I
2850   CC=0
2860 RETURN
2870 '
2880 *PASE1
2890  FOR I=1 TO N
2900   A(I,0)=A1(I) :A(0,I)=A2(I)
2910   IF P(I)=1 AND P1(I)=1 THEN P(I)=1 ELSE P(I)=P(I)+P1(I)
2920   IF Q(I)=1 AND Q1(I)=1 THEN Q(I)=1 ELSE Q(I)=Q(I)+Q1(I)
2930  NEXT I
2940   C=C+C1
2950 RETURN
2960 '
2970 *PASE2
2980  FOR I=1 TO N
2990   A(I,0)=AI(I) :A(0,I)=AJ(I)
3000   IF P(I)=1 AND PD(I)=1 THEN P(I)=1 ELSE P(I)=P(I)+PD(I)
3010   IF Q(I)+QD(I)=2 THEN Q(I)=1 ELSE Q(I)=Q(I)+QD(I)
3020  NEXT I
3030   C=C+CC
3040 RETURN
3050 '
3060 *ANS
3070 FOR K=1 TO N
3080  PRINT USING "(## ##)";K,J(K);
3090 NEXT K
3100  PRINT :I=I-1 :KEI=KEI+1
3110 RETURN
3120 '
3130 *HYOU
3140  FOR K = 1 TO N
3150     PRINT USING "#### 行    " ; K ;
3160     FOR J = 1 TO N
3170         PRINT A( K, J ) ;
3180     NEXT J : PRINT
3190  NEXT K
3200 '
3210 *CHECK
3220   PRINT  : PRINT
3230   PRINT "正しいですか  Y／N  ?" : INPUT A$
3240   IF A$ = "Y" OR A$ = "y" THEN RETURN
3250   IF A$ = "N" OR A$ = "n" THEN GOSUB *TEISEI : GOTO *HYOU
3260  GOTO *CHECK
3270  '
3280  *TEISEI
3290    PRINT "  何行目ですか?  " : INPUT K
3300    PRINT "  何列目ですか?  " : INPUT J
3310    PRINT USING "A(## , ##) = " ; K, J ;
3320    PRINT " 訂正したい値は?  " : INPUT A( K, J )
3330    PRINT " 訂正は終わりましたか  Y  ?"
3340       INPUT A$
3350   IF A$ = "Y" OR A$ = "y" THEN RETURN
3360   GOTO *TEISEI
```

```
3370 '
```

出力は画面にでるようになっている

出力をプリンター等にするときは

1085行に OPEN "LPT1:" FOR OUTPUT AS #1 入れ

プログラム中で PRINT *** を PRINT #1, *** にかえる。

C言語プログラムリスト

```
#include<stdio.h>
static int a[50][50],b[50][50],c[50][50],p[50],q[50],jj[50],f[50];
static int ai[50],aj[50],a1[50],a2[50],pd[50],qd[50],p1[50],q1[50];
static int d,cc,c1,m,n,v,z,z1,kai,min,mins,max;

main()
{
    int i,j;
    printf("¥n正方行列の次数を入力してください？ ");
    scanf("%d",&n); printf("¥n");
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            printf("%d行 %d列の値は？ ",i,j);
            scanf("%d",&a[i][j]);
            b[i][j]=a[i][j];
        }
        printf("¥n");
    }
    hyo();
    while(1)
    {
    /*-----------------手続 1 ---------------*/
        printf("¥n---手続 1 ---");
        for(i=1;i<=n;i++)
        {
            min=a[i][1];
            for(j=1;j<=n;j++)
            {
                if(min>a[i][j])
                    min=a[i][j];
            }
            for(j=1;j<=n;j++)
                a[i][j]-=min;
        }
    /*----------------手続 2 ---------------*/
        printf("¥n---手続 2 ---");
        for(j=1;j<=n;j++)
        {
            min=a[1][j];
            for(i=1;i<=n;i++)
            {
                if(min>a[i][j])
                    min=a[i][j];
            }
            for(i=1;i<=n;i++)
                a[i][j]-=min;
        }
    /*---------------手続 3 ---------------*/
        printf("¥n---手続 3 ---¥n");
```

```
    d=0; max=0; zero();
    for(i=1;i<=n;i++)
    {
        p[i]=0; q[i]=0;
        for(j=1;j<=n;j++)
        {
            if(a[i][j]==0)
                a[i][0]++;
        }
        if(a[i][0]>max)
            max=a[i][0];
    }
    for(j=1;j<=n;j++)
    {
        for(i=1;i<=n;i++)
        {
            if(a[i][j]==0)
                a[0][j]++;
        }
        if(a[0][j]>max)
            max=a[0][j];
    }
    v=1;hyouji();
    for(m=max;m>=1;m--)
    {
        dnyuuryoku();
        gyou();
        retu();
        for(i=1;i<=n;i++)
        {
            a1[i]=ai[i];a2[i]=aj[i];p1[i]=pd[i];q1[i]=qd[i];
        }
        c1=cc;
        dnyuuryoku();
        retu();
        gyou();
        if(c1>=cc)
            pass1();
        if(c1<cc)
            pass2();
        c1=0;
        printf("¥n線の数%d",d);
    }
/*-----------------手続4------------*/
    printf("¥n---手続4---");
    if(d==n) break;
/*-----------------手続5------------*/
    printf("¥n---手続5---");
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
            if((p[i]!=1)&&(q[j]!=1))
                {
                    mins=a[i][j]; i=n; j=n;
                }
    }
/*   線が引かれていない成分から最小数を決める        */
    for(i=1;i<=n;i++)
    {
        if(p[i]!=1)
        {
            for(j=1;j<=n;j++)
            {
                if(q[j]!=1)
                {
                    if(mins>a[i][j])
                        mins=a[i][j];
                }
```

```
            }
        }
    }
/*                      6-1             */
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
            a[i][j]-=mins;
    }
/*                      6-2             */
    for(i=1;i<=n;i++)
    {
            if(p[i]==1)
            {
                for(j=1;j<=n;j++)
                    a[i][j]+=mins;
            }
        }
    /*                      6-3             */
        for(j=1;j<=n;j++)
        {
            if(q[j]==1)
            {
                for(i=1;i<=n;i++)
                    a[i][j]+=mins;
            }
        }
    }


    /*      0のみの配列を探す           */
    printf("¥n ¥n"); v=0; hyouji();
    printf("¥n この配列の最適配列は¥n");
    i=1; jj[i]=0;
    do
    {
    loop:
        f[jj[i]]=0;
        if(jj[i]==n)
        {
            jj[i]=0; i--; goto loop;
        }
        z1=jj[i]+1;
        for(z=z1;z<=n;z++)
        {
            if((f[z]!=0)||(a[i][z]!=0))
            goto ss;
            {
                jj[i]=z; jj[i+1]=0; f[z]=1; z=n; i+=2;
            }
        ss:;
        }
            i--;
            if(i>n)
                i=ans(i);
    }
    while(i>0);
    printf("¥n%d個  解答があります.¥n",kai);
}


/*              サブルーチン            */

hyouji()
{
    int i,j;
    for(i=1;i<=n;i++)
    {
```

```
        for(j=1;j<=n;j++)
        {
            if(v==0)
                c[i][j]=b[i][j];
            else if(v==1)
                c[i][j]=a[i][j];
            printf("%4d",c[i][j]);
        }
        printf("¥n");
    }

    }

    zero()
    {
        int i;
        for(i=1;i<=n;i++)
        {
            a[i][0]=0; a[0][i]=0;
        }
    }

    gyou()
    {
        int i,j;
        for(i=1;i<=n;i++)
        {
            if(ai[i]==m)
            {
                pd[i]=1; cc++;
                for(j=1;j<=n;j++)
                {
                    if(a[i][j]==0)
                    {
                        ai[i]--; aj[j]--;
                    }
                }
            }
        }
    }

    retu()
    {
        int i,j;
        for(j=1;j<=n;j++)
        {
            if(aj[j]==m)
            {
                qd[j]=1; cc++;
                for(i=1;i<=n;i++)
                {
                    if(a[i][j]==0)
                    {
                        ai[i]--; aj[j]--;
                    }
                }
            }
        }
    }

    dnyuuryoku()
    {
        int i;
        for(i=1;i<=n;i++)
        {
            ai[i]=a[i][0]; aj[i]=a[0][i]; pd[i]=p[i]; qd[i]=q[i];
        }
        cc=0;
```

```
}

pass1()
{
    int i;
    for(i=1;i<=n;i++)
    {
        a[i][0]=a1[i]; a[0][i]=a2[i];
        if(p[i]==1 && p1[i]==1)
            p[i]=1;
        else
            p[i]+=p1[i];
        if( q[i]==1 && q1[i]==1 )
            q[i]=1;
        else
            q[i]+=q1[i];
    }
    d+=c1;
}

pass2()
{
    int i;
    for(i=1;i<=n;i++)
    {
        a[i][0]=ai[i]; a[0][i]=aj[i];
        if( p[i]==1 && pd[i]==1 )
            p[i]=1;
        else
            p[i]+=pd[i];
        if( q[i]+qd[i]==2 )
            q[i]=1;
        else
        q[i]+=qd[i];
    }
    d+=cc;
}

ans(int i)
{
    int k;
    for(k=1;k<=n;k++)                                    }
        printf("(%2d %2d)",k,jj[k]);
    printf("¥n"); i--; kai++;
    return(i);
}

hyo()
{
    int i,j;
    char *f;
    hyo:
    for(i=1;i<=n;i++)
    {
        printf("¥n%4d 行 ",i);
        for(j=1;j<=n;j++)
            printf(" %d",a[i][j]);
    }
    while(1)
    {
    printf("¥n¥n正しいですか。ｙ／ｎ？ ");
    f=getche();
    if(f=='y' || f=='Y')
        break;
    else if(f=='n' || f=='N')
    {
        teisei();
        goto hyo;
```

```
    }
    }
}

teisei()
{
    int i,j,k;
    char *f;
    while(1)
    {
        printf("¥n何行目ですか？");
        scanf("%d",&i);
        printf("¥n何列目ですか？");
        scanf("%d",&j);
        printf("¥nA(%2d,%2d)=%d",i,j,a[i][j]);
        printf("¥n訂正したい値は？");
        scanf("%d",&k);
        a[i][j]=k;
        printf("¥n訂正は終わりましたか？");
        f=getche();
        printf("¥n");
        if(f=='y'|| f=='Y')
            break;
    }
}
```