

第3部：『魔方陣の新研究』： 摂田 寛二

第2章： 四方陣の代数的研究（新）

第3節： 完全四方陣の研究（新）

0. 「完全四方陣」とは、前節の最後に復活採用した古称で、17の補数8ペアが全て汎対角線上に並ぶタイプの四方陣のことを言う。

各汎対角線が、 $a+b+a+b=(a+a)+(b+b)=17+17=34$ というように2ペアの補数によって成り立つので、必然的に汎対角線定和が成り立つ。

つまり、完全四方陣は、汎四方陣である。

1. さっそく次のような定義とプログラムを与えて、コンピュータに完全四方陣の解を計数させてみよう。汎四方陣の場合とは定義式やプログラムが違うが、果たして同じ解集合を作り出すだろうか。

補数定和の定義があると、一度に2数ずつを決定できるから、プログラムの長さは半分になり、無駄が省ける分、実行スピードは倍になる。

```
/** Complete Magic Squares of Order 4 */
/** File Name: 'MS44Cmplt.c' built by */
/** Kanji Setsuda on MacOS Xcode 1.5 */
/*
14 15 16 13 14 15 16 13 14 15
 2  3  4 | 1| 2| 3| 4| 1 2 3
 6  7  8 | 5| 6| 7| 8| 5 6 7
10 11 12 | 9|10|11|12| 9 10 11
14 15 16 |13|14|15|16|13 14 15
 2  3  4 | 1| 2| 3| 4| 1 2 3

[Basic Conditions]          |[Complete Conditions]
n1+n2+n3+n4=LSM           ... (1); | n1+n11=PSM           ... (9);
n5+n6+n7+n8=LSM           ... (2); | n2+n12=PSM           .. (10);
n9+n10+n11+n12=LSM        ... (3); | n3+n9=PSM            .. (11);
n13+n14+n15+n16=LSM       ... (4); | n4+n10=PSM           .. (12);
n1+n5+n9+n13=LSM          ... (5); | n5+n15=PSM           .. (13);
n2+n6+n10+n14=LSM         ... (6); | n6+n16=PSM           .. (14);
n3+n7+n11+n15=LSM         ... (7); | n7+n13=PSM           .. (15);
n4+n8+n12+n16=LSM         ... (8); | n8+n14=PSM           .. (16);
*/
/**/
#include <stdio.h>
/**/
short LSM, PSM;
short cnt, cnt2;
short nm[17], uflg[17];
short anstbl[7][17];
/**/
/* Main Program */
int main(){
short n;
printf("\n** 'Complete' Magic Squares of Order 4 **\n");
printf("** List of the 48 Standard Solutions **\n");
for(n=0;n<17;n++){nm[n]=0; uflg[n]=0;}
```

```

LSM=34; PSM=17; cnt=0; cnt3=0;
stp01(); /* Begin the Calculations */
if(cnt3>0){printans(cnt3);}
printf(" [Count = %d]\n",cnt);
printf(" OK!\n");
return 0;
}
/* Begin the Calculations */
/* Set n1 & n11=PSM-n1 */
void stp01(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[1]=a; nm[11]=b;
uflg[a]=1; uflg[b]=1;
stp02();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set n2 & n12=PSM-n2 */
void stp02(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[2]=a; nm[12]=b;
uflg[a]=1; uflg[b]=1;
stp03();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set n3 & n9=PSM-n3 */
void stp03(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[3]=a; nm[9]=b;
uflg[a]=1; uflg[b]=1;
stp04();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set n4=LSM-n1-n2-n3 & n10=PSM-n4 & n1<n4 */
void stp04(){
short a,b;
a=LSM-nm[1]-nm[2]-nm[3];
if((nm[1]<a)&&(a<17)){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[4]=a; nm[10]=b;
uflg[a]=1; uflg[b]=1;
stp05();
uflg[b]=0; uflg[a]=0;}}
}
/* Set n5 & n15=PSM-n5 */
void stp05(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[5]=a; nm[15]=b;
uflg[a]=1; uflg[b]=1;
}
}
}

```

```

        stp06();
        uflg[b]=0; uflg[a]=0;}
    }
}
/* Set n6 & n16=PSM-n6 & n1<n16 */
void stp06(){
    short a,b;
    for(a=1;a<17;a++){b=PSM-a;
        if((b>nm[1])&&(uflg[a]==0)&&(uflg[b]==0)){
            nm[6]=a; nm[16]=b;
            uflg[a]=1; uflg[b]=1;
            stp07();
            uflg[b]=0; uflg[a]=0;}
        }
}
/* Set n13=LSM-n1-n5-n9 & n7=PSM-n13 & n4<n13 */
void stp07(){
    short a,b;
    a=LSM-nm[1]-nm[5]-nm[9]; b=PSM-a;
    if((nm[4]<a)&&(a<17)){
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[13]=a; nm[7]=b;
            uflg[a]=1; uflg[b]=1;
            stp08();
            uflg[b]=0; uflg[a]=0;}}
}
/* Set n8=LSM-n5-n6-n7 & n14=PSM-n8 */
void stp08(){
    short a,b,c;
    a=LSM-nm[5]-nm[6]-nm[7]; b=PSM-a;
    c=LSM-nm[2]-nm[6]-nm[10];
    if((0<a)&&(a<17)&&(b==c)){
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[8]=a; nm[14]=b;
            uflg[a]=1; uflg[b]=1;
            ansprint();
            uflg[b]=0; uflg[a]=0;}}
}
/**/
/* Print the Answers */
void ansprint(){
    short n;
    cnt++;
    anm[cnt3][0]=cnt; anm[cnt3+1][0]=0;
    for(n=1;n<17;n++){anm[cnt3][n]=nm[n]; anm[cnt3+1][n]=0;}
    cnt3++; if(cnt3==6){printans(6); cnt3=0;}
}
/* Print the Answers */
void printans(short x){
    short l,l4,m,n;
    for(m=0;m<x;m++){printf("%12d/",anm[m][0]);}
    printf("\n");
    for(l=0;l<4;l++){l4=l*4;
        for(m=0;m<x;m++){printf(" ");
            for(n=1;n<5;n++){printf("%3d",anm[m][l4+n]);}}
        printf("\n");}
}
/**/

```

* 完全四方陣の標準解48個のリスト (全) *

1/	2/	3/	4/	5/	6/
1 8 11 14	1 8 13 12	1 8 13 12	1 12 7 14	1 12 13 8	1 12 13 8
12 13 2 7	14 11 2 7	15 10 3 6	8 13 2 11	14 7 2 11	15 6 3 10
6 3 16 9	4 5 16 9	4 5 16 9	10 3 16 5	4 9 16 5	4 9 16 5
15 10 5 4	15 10 3 6	14 11 2 7	15 6 9 4	15 6 3 10	14 7 2 11
7/	8/	9/	10/	11/	12/
1 14 7 12	1 14 11 8	1 14 11 8	1 15 6 12	1 15 10 8	1 15 10 8
8 11 2 13	12 7 2 13	15 4 5 10	8 10 3 13	12 6 3 13	14 4 5 11
10 5 16 3	6 9 16 3	6 9 16 3	11 5 16 2	7 9 16 2	7 9 16 2
15 4 9 6	15 4 5 10	12 7 2 13	14 4 9 7	14 4 5 11	12 6 3 13
13/	14/	15/	16/	17/	18/
2 7 12 13	2 7 14 11	2 7 14 11	2 11 8 13	2 11 14 7	2 11 14 7
11 14 1 8	13 12 1 8	16 9 4 5	7 14 1 12	13 8 1 12	16 5 4 9
5 4 15 10	3 6 15 10	3 6 15 10	9 4 15 6	3 10 15 6	3 10 15 6
16 9 6 3	16 9 4 5	13 12 1 8	16 5 10 3	16 5 4 9	13 8 1 12
19/	20/	21/	22/	23/	24/
2 13 8 11	2 13 12 7	2 13 12 7	2 16 5 11	2 16 9 7	2 16 9 7
7 12 1 14	11 8 1 14	16 3 6 9	7 9 4 14	11 5 4 14	13 3 6 12
9 6 15 4	5 10 15 4	5 10 15 4	12 6 15 1	8 10 15 1	8 10 15 1
16 3 10 5	16 3 6 9	11 8 1 14	13 3 10 8	13 3 6 12	11 5 4 14
25/	26/	27/	28/	29/	30/
3 6 15 10	3 6 15 10	3 10 15 6	3 10 15 6	3 13 8 10	3 13 12 6
13 12 1 8	16 9 4 5	13 8 1 12	16 5 4 9	6 12 1 15	10 8 1 15
2 7 14 11	2 7 14 11	2 11 14 7	2 11 14 7	9 7 14 4	5 11 14 4
16 9 4 5	13 12 1 8	16 5 4 9	13 8 1 12	16 2 11 5	16 2 7 9
31/	32/	33/	34/	35/	36/
3 13 12 6	3 16 5 10	3 16 9 6	3 16 9 6	4 5 16 9	4 5 16 9
16 2 7 9	6 9 4 15	10 5 4 15	13 2 7 12	14 11 2 7	15 10 3 6
5 11 14 4	12 7 14 1	8 11 14 1	8 11 14 1	1 8 13 12	1 8 13 12
10 8 1 15	13 2 11 8	13 2 7 12	10 5 4 15	15 10 3 6	14 11 2 7
37/	38/	39/	40/	41/	42/
4 9 16 5	4 9 16 5	4 14 7 9	4 14 11 5	4 14 11 5	4 15 6 9
14 7 2 11	15 6 3 10	5 11 2 16	9 7 2 16	15 1 8 10	5 10 3 16
1 12 13 8	1 12 13 8	10 8 13 3	6 12 13 3	6 12 13 3	11 8 13 2
15 6 3 10	14 7 2 11	15 1 12 6	15 1 8 10	9 7 2 16	14 1 12 7
43/	44/	45/	46/	47/	48/
4 15 10 5	4 15 10 5	5 4 15 10	5 16 3 10	6 3 16 9	6 15 4 9
9 6 3 16	14 1 8 11	16 9 6 3	4 9 6 15	15 10 5 4	3 10 5 16
7 12 13 2	7 12 13 2	2 7 12 13	14 7 12 1	1 8 11 14	13 8 11 2
14 1 8 11	9 6 3 16	11 14 1 8	11 2 13 8	12 13 2 7	12 1 14 7

[Count = 48] OK!

汎四方陣と全く同じ解集合が得られた。解の出方まで完全に一致しているから、いかにも判断しやすい。一瞬不思議な感じがすると思うが、その感覚をぜひ大切に保持してほしい。これは四方陣の場合にのみ見られる特有の現象だからだ。

2. 次に「変換法」の研究をしよう。汎四方陣の時のような行列の逐次移動による変換でも良いのだが、せっかく補数セットが揃ったのだから、対称四方陣の時のような独自の行列交換の規則をさがし出そう。完全四方陣だけのために。

下図のような「補数表現形」を用いて、チェックする。

足して17になる2数 $\{x, y\}$ を $\{x, \underline{x}\}$ と表現する。 $x + \underline{x} = 17$ である。

完全四方陣では、 $n1 + n\underline{1} = 17$; であるから $n\underline{1}$ を $n\underline{1}$;

$n6 + n\underline{6} = 17$; であるから $n\underline{6}$ を $n\underline{6}$;

$n2 + n\underline{2} = 17$; であるから $n\underline{2}$ を $n\underline{2}$; ...

と表現する。

この表現法のメリットは、全ての汎対角線の定和が、例えば、

$$\begin{aligned} n1 + n6 + n11 + n16 &= n1 + n6 + n\underline{1} + n\underline{6} = (n1 + n\underline{1}) + (n6 + n\underline{6}) = 17 + 17 = 34 \\ n1 + n8 + n11 + n14 &= n1 + n8 + n\underline{1} + n\underline{8} = (n1 + n\underline{1}) + (n8 + n\underline{8}) = 17 + 17 = 34 \\ n2 + n7 + n12 + n13 &= n2 + n7 + n\underline{2} + n\underline{7} = (n2 + n\underline{2}) + (n7 + n\underline{7}) = 17 + 17 = 34 \\ n2 + n5 + n12 + n15 &= n2 + n5 + n\underline{2} + n\underline{5} = 17 + 17 = 34 \\ n3 + n8 + n9 + n14 &= n3 + n8 + n\underline{3} + n\underline{8} = 34 \\ \dots \end{aligned}$$

というようにして面倒な計算無しで調べられることである。

下図は、最左図を原形として行列をいろいろ交換した場合の例図であるが、どの図でも汎対角線の定和が成り立つことが直視できる。

完全四方阵	上下(2-4)交換	上下(1-3)交換																																																																								
<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>4</td><td>1</td><td>2</td><td>3</td><td>4</td><td>1</td></tr> <tr><td>8</td><td>5</td><td>6</td><td>7</td><td>8</td><td>5</td></tr> <tr><td>2</td><td>3</td><td>4</td><td>1</td><td>2</td><td>3</td></tr> <tr><td>6</td><td>7</td><td>8</td><td>5</td><td>6</td><td>7</td></tr> </table>	4	1	2	3	4	1	8	5	6	7	8	5	2	3	4	1	2	3	6	7	8	5	6	7	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>4</td><td>1</td><td>2</td><td>3</td><td>4</td><td>1</td></tr> <tr><td>6</td><td>7</td><td>8</td><td>5</td><td>6</td><td>7</td></tr> <tr><td>2</td><td>3</td><td>4</td><td>1</td><td>2</td><td>3</td></tr> <tr><td>8</td><td>5</td><td>6</td><td>7</td><td>8</td><td>5</td></tr> </table>	4	1	2	3	4	1	6	7	8	5	6	7	2	3	4	1	2	3	8	5	6	7	8	5	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>2</td><td>3</td><td>4</td><td>1</td><td>2</td><td>3</td></tr> <tr><td>8</td><td>5</td><td>6</td><td>7</td><td>8</td><td>5</td></tr> <tr><td>4</td><td>1</td><td>2</td><td>3</td><td>4</td><td>1</td></tr> <tr><td>6</td><td>7</td><td>8</td><td>5</td><td>6</td><td>7</td></tr> </table>	2	3	4	1	2	3	8	5	6	7	8	5	4	1	2	3	4	1	6	7	8	5	6	7
4	1	2	3	4	1																																																																					
8	5	6	7	8	5																																																																					
2	3	4	1	2	3																																																																					
6	7	8	5	6	7																																																																					
4	1	2	3	4	1																																																																					
6	7	8	5	6	7																																																																					
2	3	4	1	2	3																																																																					
8	5	6	7	8	5																																																																					
2	3	4	1	2	3																																																																					
8	5	6	7	8	5																																																																					
4	1	2	3	4	1																																																																					
6	7	8	5	6	7																																																																					
完全四方阵	左右(2-4)交換	左右(1-3)交換																																																																								
<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>4</td><td>1</td><td>2</td><td>3</td><td>4</td><td>1</td></tr> <tr><td>8</td><td>5</td><td>6</td><td>7</td><td>8</td><td>5</td></tr> <tr><td>2</td><td>3</td><td>4</td><td>1</td><td>2</td><td>3</td></tr> <tr><td>6</td><td>7</td><td>8</td><td>5</td><td>6</td><td>7</td></tr> </table>	4	1	2	3	4	1	8	5	6	7	8	5	2	3	4	1	2	3	6	7	8	5	6	7	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>2</td><td>1</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>6</td><td>5</td><td>8</td><td>7</td><td>6</td><td>5</td></tr> <tr><td>4</td><td>3</td><td>2</td><td>1</td><td>4</td><td>3</td></tr> <tr><td>8</td><td>7</td><td>6</td><td>5</td><td>8</td><td>7</td></tr> </table>	2	1	4	3	2	1	6	5	8	7	6	5	4	3	2	1	4	3	8	7	6	5	8	7	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>4</td><td>3</td><td>2</td><td>1</td><td>4</td><td>3</td></tr> <tr><td>8</td><td>7</td><td>6</td><td>5</td><td>8</td><td>7</td></tr> <tr><td>2</td><td>1</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>6</td><td>5</td><td>8</td><td>7</td><td>6</td><td>5</td></tr> </table>	4	3	2	1	4	3	8	7	6	5	8	7	2	1	4	3	2	1	6	5	8	7	6	5
4	1	2	3	4	1																																																																					
8	5	6	7	8	5																																																																					
2	3	4	1	2	3																																																																					
6	7	8	5	6	7																																																																					
2	1	4	3	2	1																																																																					
6	5	8	7	6	5																																																																					
4	3	2	1	4	3																																																																					
8	7	6	5	8	7																																																																					
4	3	2	1	4	3																																																																					
8	7	6	5	8	7																																																																					
2	1	4	3	2	1																																																																					
6	5	8	7	6	5																																																																					
完全四方阵	(1-2)&(3-4)交換	(1-2)&(3-4)交換																																																																								
<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>4</td><td>1</td><td>2</td><td>3</td><td>4</td><td>1</td></tr> <tr><td>8</td><td>5</td><td>6</td><td>7</td><td>8</td><td>5</td></tr> <tr><td>2</td><td>3</td><td>4</td><td>1</td><td>2</td><td>3</td></tr> <tr><td>6</td><td>7</td><td>8</td><td>5</td><td>6</td><td>7</td></tr> </table>	4	1	2	3	4	1	8	5	6	7	8	5	2	3	4	1	2	3	6	7	8	5	6	7	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>8</td><td>5</td><td>6</td><td>7</td><td>8</td><td>5</td></tr> <tr><td>4</td><td>1</td><td>2</td><td>3</td><td>4</td><td>1</td></tr> <tr><td>6</td><td>7</td><td>8</td><td>5</td><td>6</td><td>7</td></tr> <tr><td>2</td><td>3</td><td>4</td><td>1</td><td>2</td><td>3</td></tr> </table>	8	5	6	7	8	5	4	1	2	3	4	1	6	7	8	5	6	7	2	3	4	1	2	3	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>3</td><td>2</td><td>1</td><td>4</td><td>3</td><td>2</td></tr> <tr><td>7</td><td>6</td><td>5</td><td>8</td><td>7</td><td>6</td></tr> <tr><td>1</td><td>4</td><td>3</td><td>2</td><td>1</td><td>4</td></tr> <tr><td>5</td><td>8</td><td>7</td><td>6</td><td>5</td><td>8</td></tr> </table>	3	2	1	4	3	2	7	6	5	8	7	6	1	4	3	2	1	4	5	8	7	6	5	8
4	1	2	3	4	1																																																																					
8	5	6	7	8	5																																																																					
2	3	4	1	2	3																																																																					
6	7	8	5	6	7																																																																					
8	5	6	7	8	5																																																																					
4	1	2	3	4	1																																																																					
6	7	8	5	6	7																																																																					
2	3	4	1	2	3																																																																					
3	2	1	4	3	2																																																																					
7	6	5	8	7	6																																																																					
1	4	3	2	1	4																																																																					
5	8	7	6	5	8																																																																					
[補数表現]	上下交換	左右交換																																																																								

いろいろ実験して調べた結果、次図のように組み合わせるのが良いとわかった。

0/は無変換の原形、1/は上下の2-4行交換、2/は上下(1-2)&(3-4)行の内外交換、3/は1/の結果に2/の内外交換を施した合成である。

4/は左右の2-4列交換、8/は左右(1-2)&(3-4)列の内外交換、12/は4/の結果に8/の内外交換を施した合成である。

5/は1/と4/の合成、6/は2/と4/の合成、7/は3/と4/の合成である。

9/は1/と8/の合成、10/は2/と8/の合成、11/は3/と8/の合成である。

13/は1/と12/の合成、14/は2/と12/の合成、15/は3/と12/の合成である。

この変換法を仮定すると、384個の原始解を24個ずつのグループに分類できる。あわせて鏡映反転や回転による変形も考慮すると、3個ずつのグループに分類することができる。

あるいは、この変換法を積極的に使えば、3個の基本解から、16個ずつ48個の代表解のリストが再構成できる。鏡映反転や回転による変形も考慮すると、さらに8倍のスケールの原

始解 384個のリストを再現することができる。

0/	1/	2/	3/
1 2 3 4	1 2 3 4	5 6 7 8	-7 -8 -5 -6
5 6 7 8	-7 -8 -5 -6	1 2 3 4	1 2 3 4
-3 -4 -1 -2	-3 -4 -1 -2	-7 -8 -5 -6	5 6 7 8
-7 -8 -5 -6	5 6 7 8	-3 -4 -1 -2	-3 -4 -1 -2
4/	5/	6/	7/
1 4 3 2	1 4 3 2	5 8 7 6	-7 -6 -5 -8
5 8 7 6	-7 -6 -5 -8	1 4 3 2	1 4 3 2
-3 -2 -1 -4	-3 -2 -1 -4	-7 -6 -5 -8	5 8 7 6
-7 -6 -5 -8	5 8 7 6	-3 -2 -1 -4	-3 -2 -1 -4
8/	9/	10/	11/
2 1 4 3	2 1 4 3	6 5 8 7	-8 -7 -6 -5
6 5 8 7	-8 -7 -6 -5	2 1 4 3	2 1 4 3
-4 -3 -2 -1	-4 -3 -2 -1	-8 -7 -6 -5	6 5 8 7
-8 -7 -6 -5	6 5 8 7	-4 -3 -2 -1	-4 -3 -2 -1
12/	13/	14/	15/
4 1 2 3	4 1 2 3	8 5 6 7	-6 -7 -8 -5
8 5 6 7	-6 -7 -8 -5	4 1 2 3	4 1 2 3
-2 -3 -4 -1	-2 -3 -4 -1	-6 -7 -8 -5	8 5 6 7
-6 -7 -8 -5	8 5 6 7	-2 -3 -4 -1	-2 -3 -4 -1

3. 本当にそんなことができるのか、早速実験してみよう。今回は、3個の基本解から16倍の48個の標準解を再構成しよう。

(1) 先ず前節で得た48個のリストを比較検討用に保存しておく。

次にその中から、次の不等式によって、基本解3個を選び出す。

$n1 < n4 < n13$; $n1 < n16$; に加えて $n1 = 1$; $n2 < n4$; $n5 < n13$

(2) 基本解3個に上図の16通りの変換を実行する。

下図が、その結果のリストである。保存したリストと比較・対照してみよう。

* 完全四角陣の基本解3個のリスト (全) *

1F/	2F/	3F/																																																
<table border="0" style="width: 100%; text-align: center;"> <tr><td>1</td><td>8</td><td>11</td><td>14</td></tr> <tr><td>12</td><td>13</td><td>2</td><td>7</td></tr> <tr><td>6</td><td>3</td><td>16</td><td>9</td></tr> <tr><td>15</td><td>10</td><td>5</td><td>4</td></tr> </table>	1	8	11	14	12	13	2	7	6	3	16	9	15	10	5	4	<table border="0" style="width: 100%; text-align: center;"> <tr><td>1</td><td>8</td><td>13</td><td>12</td></tr> <tr><td>14</td><td>11</td><td>2</td><td>7</td></tr> <tr><td>4</td><td>5</td><td>16</td><td>9</td></tr> <tr><td>15</td><td>10</td><td>3</td><td>6</td></tr> </table>	1	8	13	12	14	11	2	7	4	5	16	9	15	10	3	6	<table border="0" style="width: 100%; text-align: center;"> <tr><td>1</td><td>12</td><td>7</td><td>14</td></tr> <tr><td>8</td><td>13</td><td>2</td><td>11</td></tr> <tr><td>10</td><td>3</td><td>16</td><td>5</td></tr> <tr><td>15</td><td>6</td><td>9</td><td>4</td></tr> </table>	1	12	7	14	8	13	2	11	10	3	16	5	15	6	9	4
1	8	11	14																																															
12	13	2	7																																															
6	3	16	9																																															
15	10	5	4																																															
1	8	13	12																																															
14	11	2	7																																															
4	5	16	9																																															
15	10	3	6																																															
1	12	7	14																																															
8	13	2	11																																															
10	3	16	5																																															
15	6	9	4																																															

* Count[Fundamental/All] = 3/ 48

* 変換結果のリスト : その1 *

1/	0/	0/	0/
1 8 11 14	1 8 11 14	12 13 2 7	15 10 5 4
12 13 2 7	15 10 5 4	1 8 11 14	1 8 11 14
6 3 16 9	6 3 16 9	15 10 5 4	12 13 2 7
15 10 5 4	12 13 2 7	6 3 16 9	6 3 16 9
8/	9/	0/	0/
1 14 11 8	1 14 11 8	12 7 2 13	15 4 5 10
12 7 2 13	15 4 5 10	1 14 11 8	1 14 11 8
6 9 16 3	6 9 16 3	15 4 5 10	12 7 2 13
15 4 5 10	12 7 2 13	6 9 16 3	6 9 16 3

```

      0/      0/      0/      0/
8  1 14 11  8  1 14 11 13 12 7 2 10 15 4 5
13 12 7 2 10 15 4 5  8  1 14 11  8  1 14 11
  3  6  9 16  3  6  9 16 10 15 4 5 13 12 7 2
10 15 4 5 13 12 7 2  3  6  9 16  3  6  9 16

      0/      0/      0/      44/
14 1 8 11 14 1 8 11  7 12 13 2  4 15 10 5
  7 12 13 2  4 15 10 5 14 1 8 11 14 1 8 11
  9  6  3 16  9  6  3 16  4 15 10 5  7 12 13 2
  4 15 10 5  7 12 13 2  9  6  3 16  9  6  3 16

      2/      3/      0/      0/
  1  8 13 12  1  8 13 12 14 11 2 7 15 10 3 6
14 11 2 7 15 10 3 6  1  8 13 12  1  8 13 12
  4  5 16  9  4  5 16  9 15 10 3 6 14 11 2 7
15 10 3 6 14 11 2 7  4  5 16  9  4  5 16  9
. . . . .

```

だが、このままでは不十分だとわかる。変換解の大部分が $n1 < n4 < n13; n1 < n16;$ という標準解リスト用の出力条件に違反してしまう。変換にはおよそ付きものの副作用だ。

これでは具合が悪いので、 $n1 < n4 < n13; n1 < n16;$ の範囲に収まるようにあらためて構成解を清書・変形する。次のようなプログラムを使う。

```

/**/
/* Normalise Each Transformation */
void nrmlz(){
  short m,n;
  for(m=0;m<16;m++){
    for(n=1;n<17;n++){cnm[n]=t16ans[m][n];}
    while((cnm[1]>cnm[4])||(cnm[1]>cnm[13])||(cnm[1]>cnm[16])){rotat();};
    while(cnm[4]>cnm[13]){reflct();};
    for(n=1;n<17;n++){t16ans[m][n]=cnm[n];}
  }
}
/**/
/* Mirror Reflection */
void reflct(void){
  short n;
  dnm[1]=cnm[1];  dnm[2]=cnm[5];  dnm[3]=cnm[9];    dnm[4]=cnm[13];
  dnm[5]=cnm[2];  dnm[6]=cnm[6];  dnm[7]=cnm[10];   dnm[8]=cnm[14];
  dnm[9]=cnm[3];  dnm[10]=cnm[7]; dnm[11]=cnm[11];  dnm[12]=cnm[15];
  dnm[13]=cnm[4]; dnm[14]=cnm[8]; dnm[15]=cnm[12];  dnm[16]=cnm[16];
  for(n=1;n<17;n++){cnm[n]=dnm[n];}
}
/**/
/* Rotation by 90 degrees clockwise */
void rotat(void){
  short n;
  dnm[1]=cnm[13]; dnm[2]=cnm[9];    dnm[3]=cnm[5];  dnm[4]=cnm[1];
  dnm[5]=cnm[14]; dnm[6]=cnm[10];   dnm[7]=cnm[6];  dnm[8]=cnm[2];
  dnm[9]=cnm[15]; dnm[10]=cnm[11];  dnm[11]=cnm[7]; dnm[12]=cnm[3];
  dnm[13]=cnm[16]; dnm[14]=cnm[12]; dnm[15]=cnm[8];  dnm[16]=cnm[4];
  for(n=1;n<17;n++){cnm[n]=dnm[n];}
}
/**/

```

* 基本解3個から変換して完全四方陣の標準解48個を再構成する実験-2 *

1/ 10/ 47/ 39/
 1 8 11 14 1 15 6 12 6 3 16 9 4 14 7 9
 12 13 2 7 8 10 3 13 15 10 5 4 5 11 2 16
 6 3 16 9 11 5 16 2 1 8 11 14 10 8 13 3
 15 10 5 4 14 4 9 7 12 13 2 7 15 1 12 6

8/ 9/ 33/ 34/
 1 14 11 8 1 14 11 8 3 16 9 6 3 16 9 6
 12 7 2 13 15 4 5 10 10 5 4 15 13 2 7 12
 6 9 16 3 6 9 16 3 8 11 14 1 8 11 14 1
 15 4 5 10 12 7 2 13 13 2 7 12 10 5 4 15

45/ 22/ 13/ 29/
 5 4 15 10 2 16 5 11 2 7 12 13 3 13 8 10
 16 9 6 3 7 9 4 14 11 14 1 8 6 12 1 15
 2 7 12 13 12 6 15 1 5 4 15 10 9 7 14 4
 11 14 1 8 13 3 10 8 16 9 6 3 16 2 11 5

43/ 21/ 20/ 44/
 4 15 10 5 2 13 12 7 2 13 12 7 4 15 10 5
 9 6 3 16 16 3 6 9 11 8 1 14 14 1 8 11
 7 12 13 2 5 10 15 4 5 10 15 4 7 12 13 2
 14 1 8 11 11 8 1 14 16 3 6 9 9 6 3 16

2/ 3/ 36/ 35/
 1 8 13 12 1 8 13 12 4 5 16 9 4 5 16 9
 14 11 2 7 15 10 3 6 15 10 3 6 14 11 2 7
 4 5 16 9 4 5 16 9 1 8 13 12 1 8 13 12
 15 10 3 6 14 11 2 7 14 11 2 7 15 10 3 6

5/ 6/ 38/ 37/
 1 12 13 8 1 12 13 8 4 9 16 5 4 9 16 5
 14 7 2 11 15 6 3 10 15 6 3 10 14 7 2 11
 4 9 16 5 4 9 16 5 1 12 13 8 1 12 13 8
 15 6 3 10 14 7 2 11 14 7 2 11 15 6 3 10

26/ 15/ 14/ 25/
 3 6 15 10 2 7 14 11 2 7 14 11 3 6 15 10
 16 9 4 5 16 9 4 5 13 12 1 8 13 12 1 8
 2 7 14 11 3 6 15 10 3 6 15 10 2 7 14 11
 13 12 1 8 13 12 1 8 16 9 4 5 16 9 4 5

28/ 18/ 17/ 27/
 3 10 15 6 2 11 14 7 2 11 14 7 3 10 15 6
 16 5 4 9 16 5 4 9 13 8 1 12 13 8 1 12
 2 11 14 7 3 10 15 6 3 10 15 6 2 11 14 7
 13 8 1 12 13 8 1 12 16 5 4 9 16 5 4 9

4/ 11/ 46/ 40/
 1 12 7 14 1 15 10 8 5 16 3 10 4 14 11 5
 8 13 2 11 12 6 3 13 4 9 6 15 9 7 2 16
 10 3 16 5 7 9 16 2 14 7 12 1 6 12 13 3
 15 6 9 4 14 4 5 11 11 2 13 8 15 1 8 10

7/ 12/ 32/ 31/
 1 14 7 12 1 15 10 8 3 16 5 10 3 13 12 6
 8 11 2 13 14 4 5 11 6 9 4 15 16 2 7 9
 10 5 16 3 7 9 16 2 12 7 14 1 5 11 14 4
 15 4 9 6 12 6 3 13 13 2 11 8 10 8 1 15

	48/		23/		16/		30/								
6	15	4	9	2	16	9	7	2	11	8	13	3	13	12	6
3	10	5	16	11	5	4	14	7	14	1	12	10	8	1	15
13	8	11	2	8	10	15	1	9	4	15	6	5	11	14	4
12	1	14	7	13	3	6	12	16	5	10	3	16	2	7	9
	42/		24/		19/		41/								
4	15	6	9	2	16	9	7	2	13	8	11	4	14	11	5
5	10	3	16	13	3	6	12	7	12	1	14	15	1	8	10
11	8	13	2	8	10	15	1	9	6	15	4	6	12	13	3
14	1	12	7	11	5	4	14	16	3	10	5	9	7	2	16

* Monitor List of Correspondence *

?: 0

1: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

17: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

33: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

OK!

すでにおわかりのように、鏡映反転と回転を組み合わせて清書したのである。こうすれば、構成解を変質させずに変形できる。譬えて言えば、バラバラに散乱したカードを方向をそろえて並べ直すだけと想像していただきたい。

元のリストと比較しながら番号でモニターしたところ、48解の全てを欠落なく重複なく再現できていることがわかる。

かくして期せずして、3×16のグループ分けもできた。グループ内の16解は、それぞれお互いに良く似ている。場所こそ違え、行列の内容が全く同じ数字の組合せで出来ている。これらを本質的に「同一解」だと主張する人がいても、もう誰も驚くまい。

4. 今度は、コンピュータに頼らず手計算で、基本解3個を求める手だてを考えよう。当然のことながら、少し代数計算が必要になる。汎四方陣・完全四方陣の代数的「構造研究」を続けることにしよう。

(1) 方陣内部の縦横2行2列の隣接する正方四数の和に注目する。

例えば、{n1, n2, n5, n6} に注目して次の4式の辺々を足すと、

$$n1 + n2 + n3 + n4 = 34 \quad \dots\dots\dots (1)$$

$$n5 + n6 + n7 + n8 = 34 \quad \dots\dots\dots (2)$$

$$n1 + n5 + n9 + n13 = 34 \quad \dots\dots\dots (5)$$

$$+ n2 + n6 + n10 + n14 = 34 \quad \dots\dots\dots (6)$$

$$2 * \{n1 + n2 + n5 + n6\} + (n3 + n9) + (n4 + n10) + (n7 + n13) + (n8 + n14) = 4 * 34$$

補数定和の定義式によって

$$2 * \{n1 + n2 + n5 + n6\} + 17 + 17 + 17 + 17 = 136$$

よって、 $n1 + n2 + n5 + n6 = 34$

同様に {n2, n3, n6, n7} の4数に注目して

$$n1 + n2 + n3 + n4 = 34 \quad \dots\dots\dots (1)$$

$$n5 + n6 + n7 + n8 = 34 \quad \dots\dots\dots (2)$$

$$n2 + n6 + n10 + n14 = 34 \quad \dots\dots\dots (6)$$

$$+ n3 + n7 + n11 + n15 = 34 \quad \dots\dots\dots (7)$$

$$2 * \{n2 + n3 + n6 + n7\} + (n1 + n11) + (n4 + n10) + (n5 + n15) + (n8 + n14) = 4 * 34$$

$$2 * \{n2 + n3 + n6 + n7\} + 17 + 17 + 17 + 17 = 136$$

よって、 $n2 + n3 + n6 + n7 = 34$

同様に $n3 + n4 + n7 + n8 = 34$

$$n5 + n6 + n9 + n10 = 34$$

16	13	14	15	16	13
4	1	2	3	4	1
8	5	6	7	8	5
12	9	10	11	12	9
16	13	14	15	16	13
4	1	2	3	4	1

$$\begin{aligned}
n_6 + n_7 + n_{10} + n_{11} &= 34 \\
n_7 + n_8 + n_{11} + n_{12} &= 34 \\
n_9 + n_{10} + n_{13} + n_{14} &= 34 \\
n_{10} + n_{11} + n_{14} + n_{15} &= 34 \\
n_{11} + n_{12} + n_{15} + n_{16} &= 34 \\
\text{さらに} \\
n_1 + n_2 + n_{13} + n_{14} &= 34 \\
n_1 + n_5 + n_4 + n_8 &= 34 \\
n_1 + n_4 + n_{13} + n_{16} &= 34 \\
\dots
\end{aligned}$$

も成り立つ。拡張空間の任意の縦横 2×2 正方 4 数の和が定数 34 に等しい。

こういう特徴的な構造は昔から知られていて、「相結陣」として長く珍重されて来た。小生は、これを「正方連結四数定和型」と呼んでいる。

完全四方陣は、常に同時に「正方連結型四方陣」なのである。

試みに最初から「正方連結四方陣」を目指し、定義式を増やして構成してみても、結局のところ汎四方陣や完全四方陣と同じ解集合が出来てしまう。これが、四次の魔方陣の一番不思議なところである。

5. 完全方陣では、移動変換の規則ゆえに拡張空間で任意の 4 行 4 列 4 斜を選んで新方陣を構成しても、各々の和が定数 34 に等しいという条件が成り立つ。

ということは、その拡張空間に属する同じ系列の解の全部を代表させるのに、 $n_1 = 1$ の標準形 1 つを任意に選んで良いことを意味する。

n_1 には 1 以外の任意の数値も来れるが、移動可能の規則ゆえに後で任意の内部変換が可能だ。だから、他の場所に 1 があっても、いつでもホームポジションの n_1 に戻すことができる。 $n_1 = 1$ の標準形を調べれば、同系列の方陣全部を調べたのと同じ結果になる。

よって、これからは総代解として、 $n_1 = 1$ と $n_{11} = 16$ を仮定する。

ところで、 n_1 を含む 3 式から、

$$\begin{aligned}
n_1 + n_2 + n_3 + n_4 &= 34 \quad \dots\dots (1) \\
n_1 + n_5 + n_9 + n_{13} &= 34 \quad \dots\dots (5) \\
+) n_1 + n_6 + n_{11} + n_{16} &= 34 \quad \dots\dots (9)
\end{aligned}$$

$$\{n_2 + n_4 + n_5 + n_{13}\} + 2 \cdot n_1 + (n_6 + n_{16}) + (n_3 + n_9) + (n_1 + n_{11}) = 3 \cdot 34$$

ところが、(21) 式と $n_1 = 1$ から

$$\{n_2 + n_4 + n_5 + n_{13}\} + 2 + 17 + 17 + 17 = 3 \cdot 34$$

$$\text{ゆえに } n_2 + n_4 + n_5 + n_{13} = 3 \cdot 34 - 2 - (17 + 17 + 17) = 49 \quad \dots (24)$$

同様に n_{11} を含む 3 式から、

$$\begin{aligned}
n_9 + n_{10} + n_{11} + n_{12} &= 34 \quad \dots\dots (3) \\
n_3 + n_7 + n_{11} + n_{15} &= 34 \quad \dots\dots (7) \\
+) n_1 + n_6 + n_{11} + n_{16} &= 34 \quad \dots\dots (9)
\end{aligned}$$

$$\{n_7 + n_{10} + n_{12} + n_{15}\} + (n_3 + n_9) + (n_6 + n_{16}) + (n_1 + n_{11}) + 2 \cdot n_{11} = 3 \cdot 34$$

ところが、補数定和式と $n_{11} = 16$ から

$$\{n_7 + n_{10} + n_{12} + n_{15}\} + 17 + 17 + 17 + 2 \cdot 16 = 3 \cdot 34$$

$$\text{ゆえに } n_7 + n_{10} + n_{12} + n_{15} = 19 \quad \dots (25) \text{ と決まる。}$$

n_1 に縦横に隣接する 4 数 $\{n_2, n_4, n_5, n_{13}\}$ の和は 49 でなければならない、また n_{11} に縦横に隣接する 4 数 $\{n_7, n_{10}, n_{12}, n_{15}\}$ の和は 19 でなければならない。つまり、 n_{11} の周囲には比較的小さい数が集まり、 n_1 の周囲には比較的大きな数が集まることがわかる。

ところで、(24) 式のように 4 数の和が 49 となるような重複しない自然数の組み合わせは何通りあるだろうか。先ずそのうちの 3 数を考えられる最大値に取ると、 $\{15, 14, 13, 7\}$ となる（‘組み合わせ’だから大きい順に並べる）。いまは 7 以上の 4 数の組み合わせをを考え

ればよい。あれこれ試してみると、他に

{15, 14, 12, 8}, {15, 14, 11, 9}, {15, 13, 12, 9},
 {15, 13, 11, 10}, {14, 13, 12, 11} が得られる。これで全部である。

(25)式の4数は、(24)式の各項の補数 {(n2, n12), (n4, n10), (n5, n15), (n13, n7)} になっているから、あわせて次の6通りを調べればすむことがわかる。

- [1] {15, 14, 13, 7}, {2, 3, 4, 10}
- [2] {15, 14, 12, 8}, {2, 3, 5, 9}
- [3] {15, 14, 11, 9}, {2, 3, 6, 8}
- [4] {15, 13, 12, 9}, {2, 4, 5, 8}
- [5] {15, 13, 11, 10}, {2, 4, 6, 7}
- [6] {14, 13, 12, 11}, {3, 4, 5, 6}

これで n1, n11 に続いて {n2, n4, n5, n13}, {n12, n10, n15, n7} の候補が絞れた。

後は、{n3, n9}, {n6, n16}, {n8, n14} を調べればよいわけだが、どの値も一義的に決まる。{n1, n2, n4, n5, n13} を決めさえすれば自動的に決まる。

$$\begin{aligned} n3 &= 34 - n1 - n2 - n4; & n9 &= 17 - n3 \\ n6 &= 34 - n1 - n2 - n5; & n16 &= 17 - n6 \\ n8 &= 34 - n5 - n6 - n7; & n14 &= 17 - n8 \end{aligned}$$

初めに {15, 14, 13, 7}, {2, 3, 4, 10} を調べる。

4	·	7	·	4	·	7	·					
·	14		1		15		n3		14		1	15
10	·		13		·		10		·		13	·
16	2		·		3		16		2		·	3
4	·		7		·		4		·		7	·
·	14	1	15	n3	14	1	15					

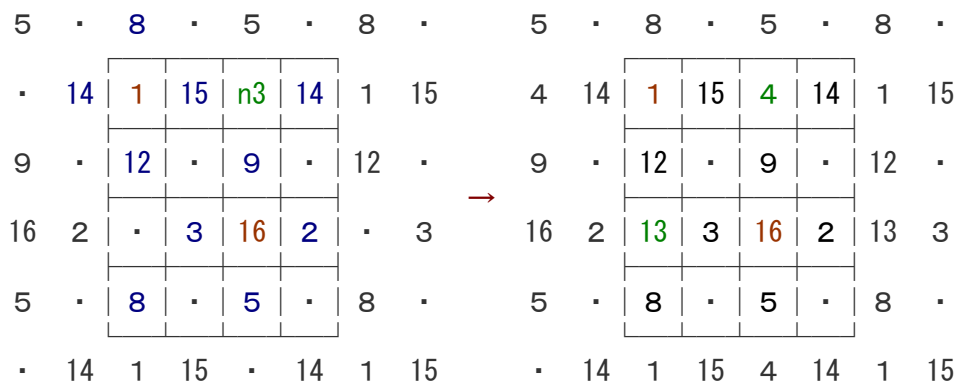
しかし、これは不可能である。n3 = 34 - 1 - 15 - 14 = 4 となり、4 が重複する。

{15, 13, 12, 9}, {2, 4, 5, 8}; {15, 13, 11, 10}, {2, 4, 6, 7} も、次々に調べる。下の左図では、n3 = 34 - 1 - 15 - 13 = 5 となり、n15 = 5 と重複する。

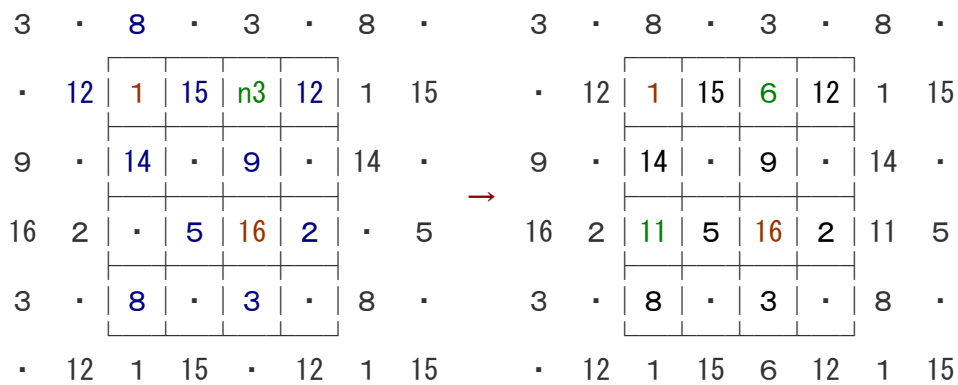
5	·	9	·	5	·	9	·	6	·	10	·	6	·	10	·										
·	13		1		15		n3		13		1	15	·	13		1		15		n3		13		1	15
8	·		12		·		8		·		12	·	7	·		11		·		7		·		11	·
16	2		·		4		16		2		·	4	16	2		n9		4		16		2		·	4
5	·		9		·		5		·		9	·	6	·		10		·		6		·		10	·
·	13	1	15	·	13	1	15	·	13	1	15	·	13	1	15										

右図は良さそうに思える。n3 = 5, n9 = 12 と決まるが、n1 + n4 + n13 + n16 = 34 が成り立たない。n16 = 34 - 1 - 13 - 10 = 10 となって、n13 = 10 と矛盾する。

次は {15, 14, 12, 8}, {2, 3, 5, 9} を調べる。



これは有望だ。残りの (6, 11), (7, 10) も何とか入りそうだ。
 では, n4 と n5 の値を入れ替えたらどうなるか。



これも有望だ。残りの (4, 13), (7, 10) も何とか入りそうだ。

解の見つけ方は何とかわかったが、正解はいったい幾つあるだろうか。全部の解を手に入
 れたいものだ。結局のところ、 $n1 = 1$, $n11 = 16$ として、

- {n2, n4, n5, n13}, {n12, n10, n15, n7} に
- [1] {15, 14, 13, 7}, {2, 3, 4, 10}
 - [2] {15, 14, 12, 8}, {2, 3, 5, 9}
 - [3] {15, 14, 11, 9}, {2, 3, 6, 8}
 - [4] {15, 13, 12, 9}, {2, 4, 5, 8}
 - [5] {15, 13, 11, 10}, {2, 4, 6, 7}
 - [6] {14, 13, 12, 11}, {3, 4, 5, 6}

を丁寧に当てはめて調べて行くしか方法は無い。

その際、 $n2, n4, n5, n13$ の大小関係はどうするか。拡張空間のいわば座標軸上にある
 4数の大小関係は予め決めておかなければならない。

大小関係を定めなくておくと、例えば $n2, n4$ の交換が起こる。そして単に左右を反転し
 ただけの図形を新方陣として1個に数えてしまう。また $n5, n13$ の交換が起こると、



図形が上下反転してしまう。 $n2, n5$ の交換が起こると、図形が 90度 (あるいは 270度) 回
 転してしまう。こうして出来た図形をそれぞれ新方陣として別個に数えるわけには行かない。

よって、ここでは、 $n2 > n5$ および $n2 > n4$, $n5 > n13$ を仮定する。

これは恣意的な仮定で、別に $n2 < n5$ および $n2 < n4$, $n5 < n13$ としてもかまわない。

ただし、一度決めた仮定は途中で変更してはいけない。

だから、 $\{n_2, n_4, n_5, n_{13}\}, \{n_{12}, n_{10}, n_{15}, n_7\}$ に

[2] $\{15, 14, 12, 8\}, \{2, 3, 5, 9\}$ と置いた時、

次に $\{15, 12, 14, 8\}, \{2, 5, 3, 9\}$ を考えるのは妥当である。

$n_2 > n_5$ および $n_2 > n_4, n_5 > n_{13}$ に違反しない。

同様に $\{15, 8, 14, 12\}, \{2, 9, 3, 5\}$ も違反しない。

つまり、 $n_2 = 15$ (最大値) と置けば、相手は $n_4 = 14, 12, 8$ の3通りが可能である。それに応じて、 n_5, n_{13} は一義的に決まる。 $n_5 > n_{13}$ としたからである。

上の [2] の時、調べる場合の数は3通りしかないことがわかった。

他の [1] ~ [6] の時も、事情は同じである。結局全部で18通り調べればよいことがわかる。そして、誰でも最大でも 15回失敗すれば、ついには正解にたどり着ける。

正解は、3個しかない。拡張空間の形式で示す。方陣枠を左右上下斜めに動かせば、あなたの作った方陣がこれら3つの中のいずれかに必ず見つかるはずである。ただし、反転しているかも知れない。回転しているかも知れない。対称移動しているかも知れない。1の周辺、16の周辺の数値を手がかりにさがしていただきたい。

*** 完全四方陣の解3個：方陣拡張空間にて示す ***

(1)

```

3 16 2 13 3 16 2 13 3 16
10 5 11 8 10 5 11 8 10 5
15 4 14 1 15 4 14 1 15 4
6 9 7 12 6 9 7 12 6 9
3 16 2 13 3 16 2 13 3 16
10 5 11 8 10 5 11 8 10 5
15 4 14 1 15 4 14 1 15 4
6 9 7 12 6 9 7 12 6 9

```

(2)

```

5 16 2 11 5 16 2 11 5 16
10 3 13 8 10 3 13 8 10 3
15 6 12 1 15 6 12 1 15 6
4 9 7 14 4 9 7 14 4 9
5 16 2 11 5 16 2 11 5 16
10 3 13 8 10 3 13 8 10 3
15 6 12 1 15 6 12 1 15 6
4 9 7 14 4 9 7 14 4 9

```

(3)

```

9 16 2 7 9 16 2 7 9 16
6 3 13 12 6 3 13 12 6 3
15 10 8 1 15 10 8 1 15 10
4 5 11 14 4 5 11 14 4 5
9 16 2 7 9 16 2 7 9 16
6 3 13 12 6 3 13 12 6 3
15 10 8 1 15 10 8 1 15 10
4 5 11 14 4 5 11 14 4 5

```

【Total Count = 3】

(May 27, 2000(初稿), January 26, 2005(改訂新稿):

MacOS Xcode 1.5 にて作表・計数: 撰田 寛二(Kanji Setsuda))

* 追加報告 *

6. 完全四方陣の3つの解を得てみて、あらためて対称四方陣の3つの解との類似性に気付かされる。行列の内容が良く似ている。何か関連があるのであろうか。

そこで不等式の立て方を共通にし、方陣の段階的な構成法を共通化・標準化すると、あらためて下図のような解を得る。2×2の正方4数で方陣を4つのブロックに分けると、ブロックごとの内容がそっくりである。これならば、対応関係を想定しても良いのではないか。

実際に次のような規則的な《変換構成法》を仮定すると、対称四方陣の解を完全四方陣の解に作り換えることができる。

(1) 縦の第3列と第4列の内容をそっくり交換する。

(2) 横の第3行と第4行の内容をそっくり交換する。

具体的なやり方を下右図に示す。

対称四方陣と完全四方陣の解の間の対応関係

《対称四方陣》 ($n1 < n16, n1 < n4 < n13,$ $n1 < n2 < n3, n1 < n5 < n9$)	《完全四方陣》 ($n1 < n11, n1 < n3 < n9,$ $n1 < n2 < n4, n1 < n5 < n13$)	《変換構成法》 原型
(1)	(1)	原型
1 14 15 4	1 14 4 15	R1 n1 n2 n3 n4
8 11 10 5	8 11 5 10	R2 n5 6 7 8
12 7 6 9	13 2 16 3	R3 n9 10 11 12
13 2 3 16	12 7 9 6	R4 n13 14 15 16
		〈C3とC4を入れ替える〉
		↓
(2)	(2)	C3 C4
1 12 15 6	1 12 6 15	n1 n2 n4 n3
8 13 10 3	8 13 3 10	n5 6 8 7
14 7 4 9	11 2 16 5	R3 n9 10 12 11
11 2 5 16	14 7 9 4	R4 n13 14 16 15
		〈R3とR4を入れ替える〉
		↓
(3)	(3)	n1 n2 n4 n3
1 12 14 7	1 12 7 14	n5 6 8 7
8 13 11 2	8 13 2 11	R3 n13 14 16 15
15 6 4 9	10 3 16 5	R4 n9 10 12 11
10 3 5 16	15 6 9 4	〈出来上がり図〉
【Count = 3】	【Count = 3】	

逆もまた可なり。この変換構成法を使うと、完全四方陣の解を用いて対称四方陣の解に作り換えることができる。

つまり、この変換構成法によって、2種類の解の間に1対1の対応関係を付けることができる。だからこそ、ともに基本解が同数ずつあるのであろう。

対称四方陣と完全四方陣の間には、まるで双児のような関連があるようだ。

(July 23, 2000(初稿) : MWCWによる計数 : 撰田 寛二 (Kanji Setsuda))

7. その後の調査で、対称型と完全型との「相互型変換の方法」が、48個の標準解集合の間では完全には成り立たないことがわかった。解の出し方をあれこれ工夫しても、最後の幾つかで一対一対応が崩れるのである。清書変形しても直せない。

一方の6解ほどが、他方の同じ相手を重複して指名してしまい、他方に相手のいない解が6個も出てしまうのである。これでは、一対一対応は完全には成り立たない。この結果には大層困惑した。この関係を前提にした様々な議論が無意味になってしまったから。

1/ 1	2/ 2	3/ 3	4/ 4	5/ 5	6/ 6
1 8 15 10	1 8 15 10	1 8 14 11	1 8 14 11	1 8 12 13	1 8 12 13
12 13 6 3	14 11 4 5	12 13 7 2	15 10 4 5	14 11 7 2	15 10 6 3
14 11 4 5	12 13 6 3	15 10 4 5	12 13 7 2	15 10 6 3	14 11 7 2
7 2 9 16	7 2 9 16	6 3 9 16	6 3 9 16	4 5 9 16	4 5 9 16
7/ 7	8/ 8	9/ 9	10/ 10	11/ 11	12/ 12
1 12 15 6	1 12 14 7	1 12 8 13	1 12 8 13	1 14 12 7	1 14 8 11
14 7 4 9	15 6 4 9	14 7 11 2	15 6 10 3	15 4 6 9	15 4 10 5
8 13 10 3	8 13 11 2	15 6 10 3	14 7 11 2	8 11 13 2	12 7 13 2
11 2 5 16	10 3 5 16	4 9 5 16	4 9 5 16	10 5 3 16	6 9 3 16
13/ 13	14/ 14	15/ 15	16/ 16	17/ 17	18/ 18
2 7 16 9	2 7 16 9	2 7 13 12	2 7 13 12	2 7 11 14	2 7 11 14
11 14 5 4	13 12 3 6	11 14 8 1	16 9 3 6	13 12 8 1	16 9 5 4
13 12 3 6	11 14 5 4	16 9 3 6	11 14 8 1	16 9 5 4	13 12 8 1
8 1 10 15	8 1 10 15	5 4 10 15	5 4 10 15	3 6 10 15	3 6 10 15
19/ 19	20/ 20	21/ 21	22/ 22	23/ 23	24/ 24
2 11 16 5	2 11 13 8	2 11 7 14	2 11 7 14	2 13 11 8	2 13 7 12
13 8 3 10	16 5 3 10	13 8 12 1	16 5 9 4	16 3 5 10	16 3 9 6
7 14 9 4	7 14 12 1	16 5 9 4	13 8 12 1	7 12 14 1	11 8 14 1
12 1 6 15	9 4 6 15	3 10 6 15	3 10 6 15	9 6 4 15	5 10 4 15
25/ 15	26/ 25	27/ 13	28/ 26	29/ 29	30/ 30
3 6 16 9	3 6 16 9	3 6 13 12	3 6 13 12	3 10 16 5	3 10 13 8
10 15 5 4	13 12 2 7	10 15 8 1	16 9 2 7	13 8 2 11	16 5 2 11
13 12 2 7	10 15 5 4	16 9 2 7	10 15 8 1	6 15 9 4	6 15 12 1
8 1 11 14	8 1 11 14	5 4 11 14	5 4 11 14	12 1 7 14	9 4 7 14
31/ 33	32/ 34	33/ 3	34/ 35	35/ 1	36/ 36
3 13 10 8	3 13 6 12	4 5 15 10	4 5 15 10	4 5 14 11	4 5 14 11
16 2 5 11	16 2 9 7	9 16 6 3	14 11 1 8	9 16 7 2	15 10 1 8
6 12 15 1	10 8 15 1	14 11 1 8	9 16 6 3	15 10 1 8	9 16 7 2
9 7 4 14	5 11 4 14	7 2 12 13	7 2 12 13	6 3 12 13	6 3 12 13
37/ 39	38/ 40	39/ 43	40/ 44	41/ 17	42/ 27
4 9 15 6	4 9 14 7	4 14 9 7	4 14 5 11	5 4 16 9	5 4 16 9
14 7 1 12	15 6 1 12	15 1 6 12	15 1 10 8	10 15 3 6	11 14 2 7
5 16 10 3	5 16 11 2	5 11 16 2	9 7 16 2	11 14 2 7	10 15 3 6
11 2 8 13	10 3 8 13	10 8 3 13	6 12 3 13	8 1 13 12	8 1 13 12
43/ 42	44/ 41	45/ 5	46/ 37	47/ 32	48/ 31
5 10 11 8	5 11 10 8	6 3 15 10	6 3 15 10	6 9 12 7	6 12 9 7
16 3 2 13	16 2 3 13	9 16 4 5	12 13 1 8	15 4 1 14	15 1 4 14
4 15 14 1	4 14 15 1	12 13 1 8	9 16 4 5	3 16 13 2	3 13 16 2
9 6 7 12	9 7 6 12	7 2 14 11	7 2 14 11	10 5 8 11	10 8 5 11

* Count = 48

* Monitor List of Correspondence *

```

??: 0
1:  2 1 2 1 2 1 1 1 1 1 1 1
13: 2 1 2 1 2 1 1 1 1 1 1 1
25: 1 1 1 0 1 1 1 1 1 1 1 1
37: 1 0 1 1 1 1 1 1 0 0 0 0

```

どうも、標準解リスト出力用の不等式の副作用が混乱の原因らしい。

思い直して、解集合のレベルを下げ、より要素数の多い「原始解の集合」で対応関係を調べてみた。これならば、リスト出力用の不等式の影響を一切受けない。

解の総数が 384個もあるので、チェックが大変だが、コンピュータにやらせる分にはどうということはない。すぐにも答えを出してくれた。

わがプログラムとその実行結果を示して、本稿を締めくくりにしよう。

```

/* Correspondence between the 2 Types of Magic Squares: */
/**/
/** 'Complete' and 'Self-Complementary' of Order 4 **/
/** 'MS44CSC384.c' by Kanji Setsuda on MacOS Xcode 1.5 **/
/*
14 15 16 13 14 15 16 13 14 15

  2  3  4 | 1 | 2 | 3 | 4 | 1  2  3
  6  7  8 | 5 | 6 | 7 | 8 | 5  6  7
 10 11 12 | 9 |10 |11 |12 | 9 10 11
 14 15 16 |13 |14 |15 |16 |13 14 15
  2  3  4 | 1 | 2 | 3 | 4 | 1  2  3

[Basic Conditions]           [Complete Conditions]
n1+n2+n3+n4=C           ... rw1; | n1+n11=CC           ... Cpt1;
n5+n6+n7+n8=C           ... rw2; | n2+n12=CC           ... Cpt2;
n9+n10+n11+n12=C        ... rw3; | n3+n9=CC            ... Cpt3;
n13+n14+n15+n16=C       ... rw4; | n4+n10=CC           ... Cpt4;
n1+n5+n9+n13=C          ... cl1; | n5+n15=CC           ... Cpt5;
n2+n6+n10+n14=C         ... cl2; | n6+n16=CC           ... Cpt6;
n3+n7+n11+n15=C         ... cl3; | n7+n13=CC           ... Cpt7;
n4+n8+n12+n16=C         ... cl4; | n8+n14=CC           ... Cpt8;
*/
/**/
#include <stdio.h>
/**/
short LSM, PSM;
short cnt;
short nm[17], uflg[17];
short tnm[17], snm[17], crsptbl[385];
short ansctbl[385][17];
short ansstbl[385][18];
/**/
/* Main Program */
int main(){
short n;
printf("\n*** Complete Magic Squares of Order 4: 384 Solutions ***\n");
for(n=0;n<17;n++){nm[n]=0; uflg[n]=0;}
LSM=34; PSM=17; cnt=0;
stpc01();
printf("* Count = %d\n",cnt);
printf("\n*** Self-Complementary Magic Squares 4*4: 384 Solutions ***\n");
for(n=0;n<17;n++){nm[n]=0; uflg[n]=0;}
for(n=0;n<385;n++){crsptbl[n]=0;}
cnt=0;
stps01();
printf("* Count = %d\n",cnt);
printf("\n* Monitor List of Correspondence *\n",cnt);
monitlst();
printf("\n OK!\n");
return 0;
}

```



```

}
/* Begin The Search */
/* Set n1 & n11=PSM-n1 */
void stpc01(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[1]=a; nm[11]=b;
uflg[a]=1; uflg[b]=1;
stpc02();
uflg[b]=0; uflg[a]=0;}}
}
}
/**/
/* Set n2 & n12=PSM-n2 */
void stpc02(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[2]=a; nm[12]=b;
uflg[a]=1; uflg[b]=1;
stpc03();
uflg[b]=0; uflg[a]=0;}}
}
}
/**/
/* Set n3 & n9=PSM-n3 */
void stpc03(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[3]=a; nm[9]=b;
uflg[a]=1; uflg[b]=1;
stpc04();
uflg[b]=0; uflg[a]=0;}}
}
}
/**/
/* Set n4=LSM-n1-n2-n3 & n10=PSM-n4 */
void stpc04(){
short a,b;
a=LSM-nm[1]-nm[2]-nm[3]; b=PSM-a;
if((0<a)&&(a<17)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[4]=a; nm[10]=b;
uflg[a]=1; uflg[b]=1;
stpc05();
uflg[b]=0; uflg[a]=0;}}}
}
}
/**/
/* Set n5 & n15=PSM-n5 */
void stpc05(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[5]=a; nm[15]=b;
uflg[a]=1; uflg[b]=1;
stpc06();
uflg[b]=0; uflg[a]=0;}}
}
}

```

```

}
}
/**/
/* Set n13=LSM-n1-n5-n9 & n7=PSM-n13 */
void stpc06(){
short a,b;
a=LSM-nm[1]-nm[5]-nm[9]; b=PSM-a;
if((0<a)&&(a<17)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[13]=a; nm[7]=b;
uflg[a]=1; uflg[b]=1;
stpc07();
uflg[b]=0; uflg[a]=0;}}
}
/**/
/* Set n6 & n16=PSM-n6 */
void stpc07(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[6]=a; nm[16]=b;
uflg[a]=1; uflg[b]=1;
stpc08();
uflg[b]=0; uflg[a]=0;}}
}
}
/**/
/* Set n8=LSM-n5-n6-n7 & n14=PSM-n8 */
void stpc08(){
short a,b,c;
a=LSM-nm[5]-nm[6]-nm[7]; b=PSM-a;
c=LSM-nm[2]-nm[6]-nm[10];
if((0<a)&&(a<17)&&(b==c)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[8]=a; nm[14]=b;
uflg[a]=1; uflg[b]=1;
anscprint();
uflg[b]=0; uflg[a]=0;}}
}
}
/**/
/* Print the Answers */
void anscprint(){
short l,m,n;
ansctbl[cnt][0]=cnt+1;
for(n=1;n<17;n++){ansctbl[cnt][n]=nm[n];}
cnt++;
if(cnt%6==0){
for(l=6;l>0;l--){
printf("%13d/",cnt-l+1);}
printf("\n");
for(m=0;m<4;m++){
for(l=6;l>0;l--){
printf(" ");
for(n=1;n<5;n++){printf("%3d",ansctbl[cnt-l][m*4+n]);}
}
printf("\n");}
}
}
}
/**/

```

```
/* Self-Complementary Magic Squares of Order 4 */
/*
```

2	3	4	1	2	3	4	1	2	3	
							5	6	7	
							9	10	11	12
							13	14	15	16
							13	14	15	16

```
[Basic Conditions]
```

```
n1+n2+n3+n4=C    ... rw1;
n5+n6+n7+n8=C    ... rw2;
n9+n10+n11+n12=C ... rw3;
n13+n14+n15+n16=C ... rw4;
n1+n5+n9+n13=C   ... cl1;
n2+n6+n10+n14=C  ... cl2;
n3+n7+n11+n15=C  ... cl3;
n4+n8+n12+n16=C  ... cl4;
```

```
[Self-Complementary Conditions]
```

```
n1+n16=CC    ... Cmp1;
n2+n15=CC    ... Cmp2;
n3+n14=CC    ... Cmp3;
n4+n13=CC    ... Cmp4;
n5+n12=CC    ... Cmp5;
n6+n11=CC    ... Cmp6;
n7+n10=CC    ... Cmp7;
n8+n9=CC     ... Cmp8;
```

```
*/
```

```
/* Begin The Search */
```

```
/* Set n1 & n16=PSM-n1 */
```

```
void stps01(){
  short a,b;
  for(a=1;a<17;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[1]=a; nm[16]=b;
      uflg[a]=1; uflg[b]=1;
      stps02();
      uflg[b]=0; uflg[a]=0;}
  }
}
```

```
/**/
```

```
/* Set n2 & n15=PSM-n2 */
```

```
void stps02(){
  short a,b;
  for(a=1;a<17;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[2]=a; nm[15]=b;
      uflg[a]=1; uflg[b]=1;
      stps03();
      uflg[b]=0; uflg[a]=0;}
  }
}
```

```
/**/
```

```
/* Set n4 & n13=PSM-n4 */
```

```
void stps03(){
  short a,b;
  for(a=1;a<17;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[4]=a; nm[13]=b;
      uflg[a]=1; uflg[b]=1;
      stps04();
      uflg[b]=0; uflg[a]=0;}
  }
}
```

```
}
```

```

/**/
/* Set n3=LSM-n1-n2-n4 & n14=PSM-n3 */
void stps04(){
short a,b;
a=LSM-nm[1]-nm[2]-nm[4]; b=PSM-a;
if((0<a)&&(a<17)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[3]=a; nm[14]=b;
uflg[a]=1; uflg[b]=1;
stps05();
uflg[b]=0; uflg[a]=0;}}
}
/**/
/* Set n5 & n12=PSM-n5 */
void stps05(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[5]=a; nm[12]=b;
uflg[a]=1; uflg[b]=1;
stps06();
uflg[b]=0; uflg[a]=0;}}
}
}
/* Set n9=LSM-n1-n5-n13
& n8=LSM-n4-n12-n16
& Check n8+n9==PSM? */
void stps06(){
short a,b;
a=LSM-nm[1]-nm[5]-nm[13];
b=LSM-nm[4]-nm[12]-nm[16];
if((0<a)&&(a<17)&&(a+b==PSM)){
if((uflg[a]==0)&&(uflg[b]==0)){
nm[9]=a; nm[8]=b;
uflg[a]=1; uflg[b]=1;
stps07();
uflg[b]=0; uflg[a]=0;}}
}
}
/**/
/* Set n6 & n11=PSM-n6 */
void stps07(){
short a,b;
for(a=1;a<17;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[6]=a; nm[11]=b;
uflg[a]=1; uflg[b]=1;
stps08();
uflg[b]=0; uflg[a]=0;}}
}
}
/**/
/* Set n7=LSM-n5-n6-n8
& n10=LSM-n2-n6-n14
& Check n7+n10==PSM? */
void stps08(){
short a,b;
a=LSM-nm[5]-nm[6]-nm[8];
b=LSM-nm[2]-nm[6]-nm[14];
if((0<a)&&(a<17)&&(a+b==PSM)){

```

```

    if((uflg[a]==0)&&(uflg[b]==0)){
        nm[7]=a; nm[10]=b;
        uflg[a]=1; uflg[b]=1;
        ansprint();
        uflg[b]=0; uflg[a]=0;}}
}
/**/
/* Print the Answers */
void ansprint(){
    short l,m,n,cd;
    ansstbl[cnt][0]=cnt+1;
    for(n=1;n<17;n++){ansstbl[cnt][n]=nm[n];}
    trnsf();
    cd=fndcrsp(); ansstbl[cnt][17]=cd; crsptbl[cd]++;
    cnt++;
    if(cnt%6==0){
        for(l=6;l>0;l--){cd=cnt-l;
            printf("%10d/%3d",cd+1,ansstbl[cd][17]);}
        printf("\n");
        for(m=0;m<4;m++){
            for(l=6;l>0;l--){
                printf(" ");
                for(n=1;n<5;n++){printf("%3d",ansstbl[cnt-l][m*4+n]);}
            }
            printf("\n");
        }
    }
}
}
/**/
/*

```

1	2	3	4	1	2	4	3
5	6	7	8	5	6	8	7
9	10	11	12	13	14	16	15
13	14	15	16	9	10	12	11

[Transformation]

```

*/
/* Transform 'Self-Complementary' type into 'Complete' one */
void trnsf(){
    tnm[1]=nm[1];   tnm[2]=nm[2];   tnm[3]=nm[4];   tnm[4]=nm[3];
    tnm[5]=nm[5];   tnm[6]=nm[6];   tnm[7]=nm[8];   tnm[8]=nm[7];
    tnm[9]=nm[13];  tnm[10]=nm[14];  tnm[11]=nm[16];  tnm[12]=nm[15];
    tnm[13]=nm[9];  tnm[14]=nm[10];  tnm[15]=nm[12];  tnm[16]=nm[11];
}
/**/
/* Find the Corresponding Solution to that */
short fndcrsp(){
    short fn,m,n,mc;
    fn=0;
    for(m=0;m<384;m++){
        mc=0;
        for(n=1;n<17;n++){
            if(ansctbl[m][n]==tnm[n]){mc++;}
            else{break;}
        }
    }
}

```

```

    if(mc==16){fn=m+1; break;}
}
return fn;
}
/**/
/* Print the Monitor List of Correspondence */
void monitlst(){
short n;
printf(" ??:%3d",crsptbl[0]);
for(n=0;n<384;n++){
    if(n%24==0){printf("\n%3d:",n+1);}
    printf("%3d",crsptbl[n+1]);
}
printf("\n");
}
/**/

```

* 完全四方陣の原始解384個のリスト (部分) *

1/	2/	3/	4/	5/	6/
1 8 10 15	1 8 10 15	1 8 11 14	1 8 11 14	1 8 13 12	1 8 13 12
12 13 3 6	14 11 5 4	12 13 2 7	15 10 5 4	14 11 2 7	15 10 3 6
7 2 16 9	7 2 16 9	6 3 16 9	6 3 16 9	4 5 16 9	4 5 16 9
14 11 5 4	12 13 3 6	15 10 5 4	12 13 2 7	15 10 3 6	14 11 2 7
7/	8/	9/	10/	11/	12/
1 12 6 15	1 12 6 15	1 12 7 14	1 12 7 14	1 12 13 8	1 12 13 8
8 13 3 10	14 7 9 4	8 13 2 11	15 6 9 4	14 7 2 11	15 6 3 10
11 2 16 5	11 2 16 5	10 3 16 5	10 3 16 5	4 9 16 5	4 9 16 5
14 7 9 4	8 13 3 10	15 6 9 4	8 13 2 11	15 6 3 10	14 7 2 11
13/	14/	15/	16/	17/	18/
1 14 4 15	1 14 4 15	1 14 7 12	1 14 7 12	1 14 11 8	1 14 11 8
8 11 5 10	12 7 9 6	8 11 2 13	15 4 9 6	12 7 2 13	15 4 5 10
13 2 16 3	13 2 16 3	10 5 16 3	10 5 16 3	6 9 16 3	6 9 16 3
12 7 9 6	8 11 5 10	15 4 9 6	8 11 2 13	15 4 5 10	12 7 2 13
19/	20/	21/	22/	23/	24/
1 15 4 14	1 15 4 14	1 15 6 12	1 15 6 12	1 15 10 8	1 15 10 8
8 10 5 11	12 6 9 7	8 10 3 13	14 4 9 7	12 6 3 13	14 4 5 11
13 3 16 2	13 3 16 2	11 5 16 2	11 5 16 2	7 9 16 2	7 9 16 2
12 6 9 7	8 10 5 11	14 4 9 7	8 10 3 13	14 4 5 11	12 6 3 13
25/	26/	27/	28/	29/	30/
2 7 9 16	2 7 9 16	2 7 12 13	2 7 12 13	2 7 14 11	2 7 14 11
11 14 4 5	13 12 6 3	11 14 1 8	16 9 6 3	13 12 1 8	16 9 4 5
8 1 15 10	8 1 15 10	5 4 15 10	5 4 15 10	3 6 15 10	3 6 15 10
13 12 6 3	11 14 4 5	16 9 6 3	11 14 1 8	16 9 4 5	13 12 1 8
31/	32/	33/	34/	35/	36/
2 11 5 16	2 11 5 16	2 11 8 13	2 11 8 13	2 11 14 7	2 11 14 7
7 14 4 9	13 8 10 3	7 14 1 12	16 5 10 3	13 8 1 12	16 5 4 9
12 1 15 6	12 1 15 6	9 4 15 6	9 4 15 6	3 10 15 6	3 10 15 6
13 8 10 3	7 14 4 9	16 5 10 3	7 14 1 12	16 5 4 9	13 8 1 12
37/	38/	39/	40/	41/	42/
2 13 3 16	2 13 3 16	2 13 8 11	2 13 8 11	2 13 12 7	2 13 12 7
7 12 6 9	11 8 10 5	7 12 1 14	16 3 10 5	11 8 1 14	16 3 6 9
14 1 15 4	14 1 15 4	9 6 15 4	9 6 15 4	5 10 15 4	5 10 15 4
11 8 10 5	7 12 6 9	16 3 10 5	7 12 1 14	16 3 6 9	11 8 1 14
43/	44/	45/	46/	47/	48/
2 16 3 13	2 16 3 13	2 16 5 11	2 16 5 11	2 16 9 7	2 16 9 7
7 9 6 12	11 5 10 8	7 9 4 14	13 3 10 8	11 5 4 14	13 3 6 12
14 4 15 1	14 4 15 1	12 6 15 1	12 6 15 1	8 10 15 1	8 10 15 1
11 5 10 8	7 9 6 12	13 3 10 8	7 9 4 14	13 3 6 12	11 5 4 14

49/ 3 6 9 16 10 15 4 5 8 1 14 11 13 12 7 2	50/ 3 6 9 16 13 12 7 2 8 1 14 11 10 15 4 5	51/ 3 6 12 13 10 15 1 8 5 4 14 11 16 9 7 2	52/ 3 6 12 13 16 9 7 2 5 4 14 11 10 15 1 8	53/ 3 6 15 10 13 12 1 8 2 7 14 11 16 9 4 5	54/ 3 6 15 10 16 9 4 5 2 7 14 11 13 12 1 8
.					
355/ 15 10 3 6 1 8 13 12 14 11 2 7 4 5 16 9	356/ 15 10 3 6 4 5 16 9 14 11 2 7 1 8 13 12	357/ 15 10 5 4 1 8 11 14 12 13 2 7 6 3 16 9	358/ 15 10 5 4 6 3 16 9 12 13 2 7 1 8 11 14	359/ 15 10 8 1 4 5 11 14 9 16 2 7 6 3 13 12	360/ 15 10 8 1 6 3 13 12 9 16 2 7 4 5 11 14
361/ 16 2 7 9 3 13 12 6 10 8 1 15 5 11 14 4	362/ 16 2 7 9 5 11 14 4 10 8 1 15 3 13 12 6	363/ 16 2 11 5 3 13 8 10 6 12 1 15 9 7 14 4	364/ 16 2 11 5 9 7 14 4 6 12 1 15 3 13 8 10	365/ 16 2 13 3 5 11 8 10 4 14 1 15 9 7 12 6	366/ 16 2 13 3 9 7 12 6 4 14 1 15 5 11 8 10
367/ 16 3 6 9 2 13 12 7 11 8 1 14 5 10 15 4	368/ 16 3 6 9 5 10 15 4 11 8 1 14 2 13 12 7	369/ 16 3 10 5 2 13 8 11 7 12 1 14 9 6 15 4	370/ 16 3 10 5 9 6 15 4 7 12 1 14 2 13 8 11	371/ 16 3 13 2 5 10 8 11 4 15 1 14 9 6 12 7	372/ 16 3 13 2 9 6 12 7 4 15 1 14 5 10 8 11
373/ 16 5 4 9 2 11 14 7 13 8 1 12 3 10 15 6	374/ 16 5 4 9 3 10 15 6 13 8 1 12 2 11 14 7	375/ 16 5 10 3 2 11 8 13 7 14 1 12 9 4 15 6	376/ 16 5 10 3 9 4 15 6 7 14 1 12 2 11 8 13	377/ 16 5 11 2 3 10 8 13 6 15 1 12 9 4 14 7	378/ 16 5 11 2 9 4 14 7 6 15 1 12 3 10 8 13
379/ 16 9 4 5 2 7 14 11 13 12 1 8 3 6 15 10	380/ 16 9 4 5 3 6 15 10 13 12 1 8 2 7 14 11	381/ 16 9 6 3 2 7 12 13 11 14 1 8 5 4 15 10	382/ 16 9 6 3 5 4 15 10 11 14 1 8 2 7 12 13	383/ 16 9 7 2 3 6 12 13 10 15 1 8 5 4 14 11	384/ 16 9 7 2 5 4 14 11 10 15 1 8 3 6 12 13

* Count = 384

* 補数対称型四方陣の原始解384個のリスト (完全型との対応番号付き) *

1/ 1 1 8 15 10 12 13 6 3 14 11 4 5 7 2 9 16	2/ 2 1 8 15 10 14 11 4 5 12 13 6 3 7 2 9 16	3/ 3 1 8 14 11 12 13 7 2 15 10 4 5 6 3 9 16	4/ 4 1 8 14 11 15 10 4 5 12 13 7 2 6 3 9 16	5/ 5 1 8 12 13 14 11 7 2 15 10 6 3 4 5 9 16	6/ 6 1 8 12 13 15 10 6 3 14 11 7 2 4 5 9 16
7/ 7 1 12 15 6 8 13 10 3 14 7 4 9 11 2 5 16	8/ 8 1 12 15 6 14 7 4 9 8 13 10 3 11 2 5 16	9/ 9 1 12 14 7 8 13 11 2 15 6 4 9 10 3 5 16	10/ 10 1 12 14 7 15 6 4 9 8 13 11 2 10 3 5 16	11/ 11 1 12 8 13 14 7 11 2 15 6 10 3 4 9 5 16	12/ 12 1 12 8 13 15 6 10 3 14 7 11 2 4 9 5 16
13/ 13 1 14 15 4 8 11 10 5 12 7 6 9 13 2 3 16	14/ 14 1 14 15 4 12 7 6 9 8 11 10 5 13 2 3 16	15/ 15 1 14 12 7 8 11 13 2 15 4 6 9 10 5 3 16	16/ 16 1 14 12 7 15 4 6 9 8 11 13 2 10 5 3 16	17/ 17 1 14 8 11 12 7 13 2 15 4 10 5 6 9 3 16	18/ 18 1 14 8 11 15 4 10 5 12 7 13 2 6 9 3 16
19/ 19 1 15 14 4 8 10 11 5 12 6 7 9 13 3 2 16	20/ 20 1 15 14 4 12 6 7 9 8 10 11 5 13 3 2 16	21/ 21 1 15 12 6 8 10 13 3 14 4 7 9 11 5 2 16	22/ 22 1 15 12 6 14 4 7 9 8 10 13 3 11 5 2 16	23/ 23 1 15 8 10 12 6 13 3 14 4 11 5 7 9 2 16	24/ 24 1 15 8 10 14 4 11 5 12 6 13 3 7 9 2 16

25: 1
 49: 1
 73: 1
 97: 1
 121: 1
 145: 1
 169: 1
 193: 1
 217: 1
 241: 1
 265: 1
 289: 1
 313: 1
 337: 1
 361: 1

OK!

結論を言えば、要するに、原始解集合の間では完璧な一対一対応が成り立つのである。めでたし、めでたしと、胸を撫で下ろした。

変換法の研究は、リスト出力用の不等式の影響を受けない原始解集合の間で行うのが、どうやら鉄則のようだ。

(January 26, 2005(改訂新稿): MacOS Xcode 1.5 にて作表・計数:
 著者: 摂田 寛二(Kanji Setsuda) : JAG12001@nifty.ne.jp)
