مربع‌هاي جادويي

پدارم حياتي
دانشجوي رشته كارشناسي مهندسي فن‌آوري اطلاعات

عنوان مقاله:
مربع‌هاي جادويي

نويسنده:
پدارم حياتي
دانشجوي رشته كارشناسي مهندسي فن‌آوري اطلاعات
مركز تحصيلات تكميلي علوم پايه زنجان
pedram@iasbs.ac.ir

تاريخ تنظيم:
تير ماه ۱۳۸۴

اين مقاله را تقديم مي كنم به استاد ارجمند جناب

دكتر خرسند

تير ۱۳۸۴

فهرست مطالب

مقدمه

اين مقاله به شرح و ارائه راحل‌هاي ايجاد مربع‌هاي جادويي مي‌پردازد.
ابتدا به شرح داستان تاريخچه ايجاد مربع جادويي كه از ۳۰۰۰ سال پيش ايجاد شده است مي پردازد سپس سراغ تعريف مربع‌جادويي مي رود. در ادامه مقاله به بحث و بررسي انواع راحل‌هايي ممكن در ساخت مربع‌هاي جادويي مي پردازد و چندين مثال از مربع‌هاي جادويي موجود ارائه مي‌دهد. در پايان الگوريتم‌هاي حل چند مدل مربع‌جادويي آورده شده است.

تاريخچه

مربع‌هاي جادويي بالغ بر ۳۰۰۰ هزار سال مورد مطالعه و بحث رياضيدانان بوده‌اند. اولين مربع‌جادويي باقي‌مانده متعلق به تمدن چيني ها در ۲۲۰۰ سال پيش از ميلاد مسيح است.
در قرن نهم منجمان عرب از آن براي محاسبه جداول ساعات روزانه استفاده كردند. در ۱۳۰۰ سال پس از ميلاد مسيح مربع‌هاي جادويي به سمت غرب روانه شد. يك كتبه حكاكي شده ازيم هنرمند آلماني (Albrecht Dürer) موجود است كه نشان‌دهنده اولين نشانه‌هاي پيدايش مربع‌هاي جادويي در سرزمين غرب مي باشد. اين كتبه حاوي يك مربع‌جادويي است كه در آن تاريخ ۱۵۱۴ به صورت دو رديف عدد پي‌درپي است.
از آنجايي كه مفهوم مربع‌هاي جادويي به راحتي قابل فهم نمي باشد از آن‌ها در جدول ها و ماشين‌هاي رياضي استفاده مي‌كنند. بطوري كه آثار باقي‌مانده از بنيامين فرانكلين[1] نشان مي‌دهد كه روي يك مربع جادويي ۸×۸ بصورت تفريحي كار كرده است.

[1] Benjamin Franklin

مربع‌جادويي

آقاي آلن ادلر[1] تعريف مربع‌جادويي ساده[2] را به شرح زير داده است:
"مربع‌جادويي ترتيبي از اعداد متشكل از **1, 2, 3, ... n^2** در يك ماتريس **nxn** مـي باشـد بطـوري كه هر عدد تنها يكبار آمده باشد و مجموع هر سطر، ستون و قطر اصلي يكي باشد."
براي مثال يك مربع جادويي ۳x۳ را در زير مشاهده مي‌كنيد:

6	1	8
7	5	3
2	9	4

مجموع هر سطر، ستون و قطر اصلي آن ۱۵ مي‌باشد.

[1] Allan Adler
[2] مراد از عبارت مربع‌جادويي در اين مقاله مربع‌هاي جادويي ساده مي باشد.

فرمول محاسبه

فرض كنيد M مجموع اعداد روي هر سطر، ستون يا قطر اصلي مربع‌جادويي nxn مي باشد بدين ترتيب داريم:

$$1 + 2 + 3 + \ldots + n^2 = n \cdot M \ .$$

$$\sum_{i=1}^{n^2} i = n \cdot M$$

كه در آن مجموع تمامي اعداد موجود در مربع جادويي برابر $n \cdot M$ مي باشد پس:

$$n \cdot M = \frac{n^2(n^2+1)}{2}$$

در نتيجه مجونع اعداد هرسطر،ستون و يا قطر اصلي در يك مربع‌جادويي nxn برابر است با:

$$M = \frac{n(n^2+1)}{2}$$

مربع‌هاي جادويي

براي نمونه در زير شكل يك مربع جادويي ۳x۳ آورده شده است كه ابتدا با استفاده از فرمول فوق مجموع اعداد هر سطر، ستون و يا قطر اصلي را محاسبه مي‌كنيم، سپس عدد حاصله را با مجموع اعداد موجود روي سطرها، ستون‌ها و قطر اصلي مربع‌جادويي ۳x۳ مقايسه مي‌كنيم.

مربع‌جادويي ۳x۳:

6	1	8
7	5	3
2	9	4

$$M = \frac{3(3^2+1)}{2} = \frac{30}{2} = 15$$

مجموع سطرها:
سطر اول:
6 + 1 + 8 = 15
سطر دوم:
7 + 5 + 3 = 15
سطر سوم:
2 + 9 + 4 = 15

مجموع ستون‌ها:
ستون اول:
6 + 7 + 2 = 15
ستون دوم:
1 + 5 + 3 = 15
ستون سوم:
8 + 3 + 4 = 15

مجموع قطر اصلي:
6 + ۵ + ۴ = 15

مربع‌هاي جادويي مرتبه فرد

يك مربع‌جادويي ساده از اعداد $1, 2, 3, \ldots n^2$ استفاده مي كند. كلمه ساده كه پس از عبارت مربع‌جادويي آورده شده است بدين معناست كه در اين مربع هاي جادويي تنها از اعداد صحيح مثبت استفاده شده است.
مسئله بسيار جالبي كه وجود دارد نحوه ايجاد يك مربع‌جادويي مي باشد. براي ساخت مربع‌هاي جادويي مرتبه فرد (مربع‌جادويي مرتبه زوج مربعي است كه در آن n عددي فرد است) الگوريتم‌هاي ساده اي موجود است كه مي توان با استفاده از آن‌ها تمامي مربع‌هاي جادويي از مرتبه فرد را ساخت. بعضي از اين الگوريتم‌ها عبارتند از:

"de la Loubere's algorithm", "the staircase method", "Siamese method".

الگوريتم تشكيل يك مربع‌جادويي ۳x۳ در ادامه آورده شده است.

الگوريتم de la Loubere's:

۱. يك جدول n در n رسم مي‌كنيم.
۲. عدد يك را در ستون مياني رديف اول جدول قرار مي‌دهيم.

۳. عدد بعدي را در مسير "بالا – راست قطري" عدد قبلي قرار با توجه به موارد زير مي دهيم:
الف) اگر در حركت مسير "بالا – راست قطري" از محدوده ي جدول خارج شديم عدد جديد را در نقطه مقابل ستوني سطري، سطر و ستوني كه بايد عدد جديد را شامل شود قرار مي دهيم.
ب) اگر در حركت مسير "بالا – راست قطري" به خانه‌اي در جدول برخورديم كه از قبل پر شده بود، عدد جديد را در رديف زيرين عدد قبلي قرار مي‌دهيم.
۴. مرحله ۳ را آنقدر تكرار مي كنيم تا تمامي n^2 عدد در جاي خود قرار گيرند.
۵. پايان الگوريتم.

براي روشن شدن مراحل الگوريتم فوق شكل هاي زير را دنبال كنيد:

	1	

	1	
		2

	1	
3		
		2

	1	
3		
4		2

	1	
3	5	
4		2

	1	6
3	5	
4		2

	1	6
3	5	7
4		2

8	1	6
3	5	7
4		2

8	1	6
3	5	7
4	9	2

قبل از خواند ادامه مقاله با استفاده از اين الگوريتم يك مربع‌جادويي ۵x۵ رسم كنيد.

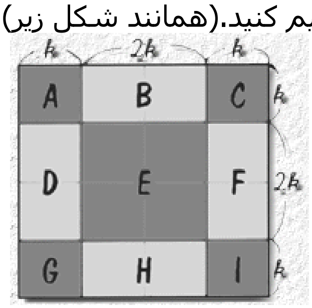
مربع‌هاي جادويي مرتبه زوج

در حال حاضر هيچ الگوريتم شناخته شده‌اي براي ساخت مربع‌هاي جادويي مرتبه زوج موجود نمي باشد(مربع‌جادويي مرتبه زوج مربعي است كه در آن n عددي زوج است)، اما چندين روش براي ساخت اين نوع از مربع‌هاي جادويي موجود است كه كلي نيست و به صورت موردي به ساخت بعضي از انواع مربع‌هاي جادويي مرتبه زوج مي‌پردازد. در ادامه به شرح انواع اين روش ها مي پردازيم.

روش ساخت مربع‌هاي جادويي سري ۴:

n=4,8,12,...

۱. مربع را به ۹ قسمت تقسيم كنيد.(همانند شكل زير)



۲. اعداد را به ترتيب از خانه سطر ۱ ستون ۱ مربع در قسمت هاي تيره رنگ(A, C, E, G, I) بنويسيد.

1	[_]	[_]	4
[_]	6	7	[_]
[_]	10	11	[_]
13	[_]	[_]	16

۳. اعداد باقي مانده رو به صورت برعكس و از راست به چپ در قسمت هاي خالي مربع قرار دهيد.

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

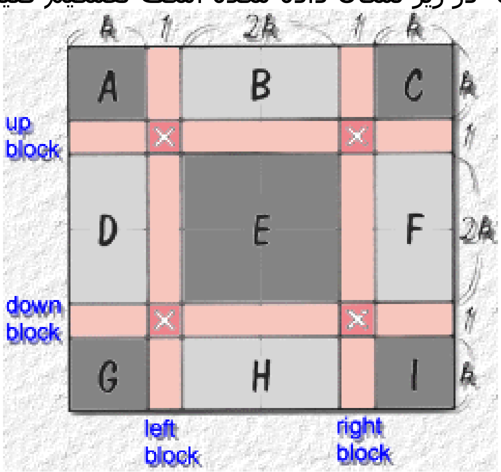
براي مثال مربع‌جادويي ۸x۸ با استفاده از اين روش به صورت زير است:

1	2	62	61	60	59	7	8
9	10	54	53	52	51	15	16
48	47	19	20	21	22	42	41
40	39	27	28	29	30	34	33
32	31	35	36	37	38	26	25
24	23	43	44	45	46	18	17
49	50	14	13	12	11	55	56
57	58	6	5	4	3	63	64

روش ساخت مربع‌هاي جادويي سري ۶:

n = 6,10,14,...

۱. مربع را به شكلي كه در زير نشان داده شده است تقسيم كنيد:



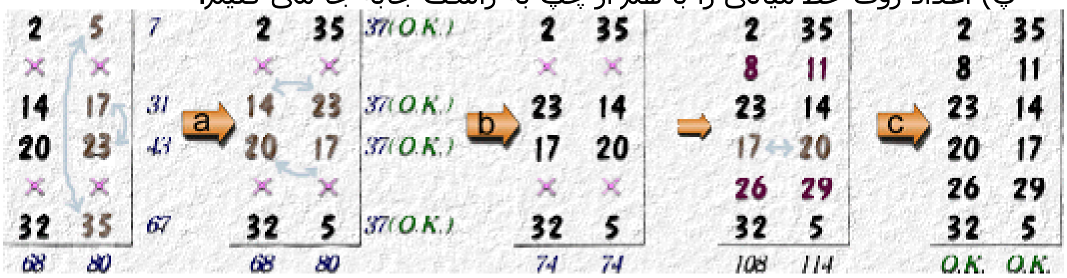
۲. همانند روش حل مربع‌هاي جادويي سري ۴ قسمت هاي تيره رنگ(A, C, E, G, I) را به ترتيب اعداد پر كنيد.
۳. در قسمت هاي سفيد رنگ (B, D, F, H) اعداد را بصورت برعكس و از راست به چپ قرار مى دهيم.

1		34	33		6	37
	×			×		
24		15	16		19	37
18		21	22		13	37
	×			×		
31		4	3		36	37
37		37	37		37	

۴. باقى مانده اعداد را روى نوارها صورتى رنگ به ترتيب بنويسيد. توجه داشته باشيد كه اين اعداد در مرحله بعدى تغيير جا مى دهند و همچنين اعداد موجود در قسمت هايى كه با علامت × مشخص شده اند ثابت و بدون جابه جايى باقى مى مانند(اعداد روى قطر اصلى).

	2			5	
7	×	9	10	×	12
	14			17	
	20			23	
25	×	27	28	×	30
	32			35	

۵. در اين مرحله به جابجايى اعداد روى نوارها مى پردازيم:
الف) اعداد سمت راست را به صورت برعكس مى چينيم.
ب) اعداد موجود روى نوار ميانى را با هم از چپ به راست جابه جا مى كنيم.
پ) اعداد روى خط ميانى را با هم از چپ به راست جابه جا مى كنيم.



۶. اگر مرحله ۵ را براى اعداد قسمت هاى بالايى و پايينى انجام داده ايد در اين حال يک جفت عدد براى خلاصى از تفاوت ميان آن ها در بند "پ" مرحله ۵ پيدا نمى كنيد بنابر اين به ترتيب زير براى اين مناطق عمل مى كنيم: يک رديف از ناحيه هاى B يا H انتخاب مى كنيم و اعداد روى آن ها را بصورت عكس تعويض مى كنيم – يک رديف از ناحيه هاى D يا F انتخاب مى كنيم و بصورت عكس تعويض مى كنيم.

1	2	33	34	35	6	O.K.
	8			11		
19	23	15	16	14	24	O.K.
18	20	21	22	17	13	O.K.
	26			29		
31	32	4	3	5	36	O.K.
42	O.K.	38	36	O.K.	32	

۷. اعداد روى رديف نوارهاى صورتى رنگ را در اين مرحله جابه جا مى كنيم: الف) اعداد هر بخش رديف را باهم جابه جا مى كنيم(از راست به چپ). ب) اعداد ميانى نوار را از بالا به پايين جابه جا مى كنيم. پ) اعداد اول هر رديف را از بالا به پايين جابه جا مى كنيم.

۸. در نهایت مربع جادویی ما به شکل زیر در می آید.

1	2	33	34	35	6
30	8	28	27	11	7
19	23	15	16	14	24
18	20	21	22	17	13
12	26	10	9	29	25
31	32	4	3	5	36

برای مثال به شکلی یک مربع جادویی ۱۰×۱۰ که با استفاده از این روش ساخته شده است توجه کنید:

1	2	3	94	95	96	97	98	9	10
11	12	13	87	86	85	84	88	19	20
80	29	23	77	76	75	74	28	22	21
70	69	68	34	35	36	37	33	62	61
60	59	58	44	45	46	47	43	52	51
41	42	53	54	55	56	57	48	49	50
40	39	38	64	65	66	67	63	32	31
30	79	73	27	26	25	24	78	72	71
81	82	83	17	16	15	14	18	89	90
91	92	93	7	6	5	4	8	99	100

تعداد مربع های جادویی

توجه داشته باشید که با چرخش یا معکوس کردن هر مربع جادویی یک مربع جادویی جدید ایجاد می شود، اما آن ها را به عنوان یک مربع جادویی متفاوت به حساب نمی آوریم، بدین ترتیب تعداد مربع های جادویی ۳×۳ برابر یک؛ ۴×۴ برابر ۸۸۰؛ ۵×۵ برابر ۱۳۰۰۰۰۰۰ می باشد! همانطور که مشاهده می کنید این تعداد بصورت اعجاب آوری زیاد می شود.
در این شمارش فقط مربع های جادویی ساده محاسبه شده اند.

برنامه کامپیوتری ساخت مربع های جادویی

```
#define SIZE 160    /* Consider a memory for increasing the size. */

#include "stdio.h"
#include "stdlib.h"
#include "conio.h"

int m[SIZE][SIZE];

void odd_num(int n);
void even_num(int n);
void output(int n);
void _swap(int i1, int j1, int i2, int j2);

main()
{
    int i,j,n;
    char *s="           ";


printf("\n*********************************************************");
```

```
printf("\n*        <  < <   MAGIC  SQUARE   > >  >                *");
printf("\n*   by;  Kwon Young Shin(brainstm@chollian.net)        *");
printf("\n*******************************************************");

    while(1) {
        printf("\n\n\n:-) Input the number(3-%d): ",SIZE);
        gets(s);
        n=atoi(s);
            if(n < 3 || n > SIZE) break;
        if(n%2) odd_num(n);
        else even_num(n);
        output(n);
    }

    printf("\n\n* .. Quit .. .  [[ <   MAGIC  SQUARE   > ]]  . ..
*\n\n\n");
}

void odd_num(int n)
{
    int i,j,num=1;
    int nn=n*3/2;

    for(i=0; i < n; i++)
        for(j=0; j < n; j++)
            m[(j-i+nn)%n][(i*2-j+n)%n]=num++;
}

void even_num(int n)
{
    int i,j,num=1;
    int nminus=n-1,nmiddle=n/2,nn=n*n+1;
    int osl=0;
    int switch_row[2];
    int last_switch_column;
    int first_block=(n-2)/4,second_block=nminus-first_block;
    int first_inside=n/4,second_inside=nminus-first_inside;

    for(j=0; j < n; j++)
        for(i=0; i < n; i++) {
            if(i  >=  first_inside  &&  i  <=  second_inside  &&  j  >=
first_inside && j <= second_inside)
                m[i][j]=num;
            else  if((i  >  first_block  &&  i  <  second_block)  ||  (j  >
first_block && j < second_block))
                m[i][j]=nn-num;
            else m[i][j]=num;
            num++;
        }
    if(!(n%4)) return;

    switch_row[0]=random(nmiddle-1)+first_block+1;
    switch_row[1]=random(nmiddle-1);
    if(switch_row[1] >= first_block) switch_row[1]+=(nmiddle+1);
    last_switch_column=random(nmiddle-1);
    if(last_switch_column               >=               first_block)
last_switch_column+=(nmiddle+1);

/* Simply, you can write as follows..
    switch_row[0]=nmiddle;
    switch_row[1]=0;
```

```
    last_switch_column=0;
*/

    for(i=0; i < nmiddle; i++) {
        if(i==first_block || i==second_block) {
            osl=1-osl;
            continue;
        }
        _swap(second_block, i, second_block, nminus-i);
        _swap(i, first_block, nminus-i, first_block);
        _swap(i, second_block, nminus-i, second_block);
        _swap(i, switch_row[osl], nminus-i, switch_row[osl]);
    }
    for(i=first_block+1; i < second_block; i++) {
        _swap(first_block, i, second_block, i);
        _swap(i, first_block, i, second_block);
    }
    _swap(first_block, nmiddle, second_block, nmiddle);
    _swap(last_switch_column,     first_block,     last_switch_column,
second_block);
}

void output(int n)
{
    int i,j,ch,err=0;
    unsigned long sum,sl,sc,sd1=0,sd2=0;

    sum=(unsigned long)n*(n*n+1)/2;
    printf(";-) SUM = %lu.     * ..check sum... .   .    ",sum);
    for(j=0; j < n; j++) {
        sd1+=m[j][j];
        sd2+=m[j][n-j-1];
        sl=0;
        sc=0;
        for(i=0; i < n; i++) {
            sl+=m[i][j];
            sc+=m[j][i];
        }
        if(sl!=sum) {
            err++;
            printf("\n>   Sum   of   the   row-(%d)   is   %lu.   It's
incorrect..",j+1,sl);
        }
        if(sc!=sum) {
            err++;
            printf("\n>   Sum   of   the   column-(%d)   is   %lu.   It's
incorrect..",j+1,sc);
        }
    }
    if(sd1!=sum) {
        err++;
        printf("\n>   Sum   of   the   diagonal-(\\)   is   %lu.   It's
incorrect..",j+1,sd1);
    }
    if(sd2!=sum) {
        err++;
        printf("\n>   Sum   of   the   diagonal-(/)   is   %lu.   It's
incorrect..",j+1,sd2);
    }
    if(err) printf("\n\n* %d errors are happened.",err);
    else printf("- O.K. -");
```

```
    printf("\n\n:-? Do you want to display(Y/n)? ");
    ch=getch();
    if(ch=='n' || ch=='N') return;
    printf("\n---+---------------------------------------------------
------------------------");
    for(j=0; j < n; j++) {
        printf("\n%2d)] ",j+1);
        for(i=0; i < n; i++)
            printf("%4d,",m[i][j]);
    }
    printf("\n---+---------------------------------------------------
------------------------");
}

void _swap(int i1, int j1, int i2, int j2)
{
    int k;

    k=m[i1][j1];
    m[i1][j1]=m[i2][j2];
    m[i2][j2]=k;
}
```

چکیده

مربع های جادویی به عنوان یکی از جالب ترین مسائل از ۳۰۰ هزار سال پیش از میلاد مسیح مورد توجه ریاضیدانان بوده است. برای ساخت مربع های جادویی مرتبه فرد الگوریتم های مشخصی وجود دارد اما برای ساخت مربع های جادویی مرتبه زوج الگوریتم مشخصی تا به حال بوجود نیامده است و با انواع روش های مختلف میتوان بصورت موردی بعضی از آن ها را ساخت. بر این اساس نوشتن یک الگوریتم یا برنامه کامپیوتری کلی برای ساخت تمامی مربع های جادویی از هر مرتبه ممکن نیست. تعداد مربع های جادویی بسیار زیاد است و با جابه جایی اعداد آن می توان مربع جدیدی را بدست آورد.
نکته آخر در این مقاله تنها به بررسی مربع های جادویی ساده که فقط از اعداد صحیح مثبت ساخته می شدند پرداخته شد حال اگر باقی مربع های جادویی را بررسی کنیم تعداد، الگوریتم ها، ... آن ها چه خواهد شد؟!

منابع

- ✓ http://www.jcu.edu/math/vignettes/index.html
- ✓ http://forum.swarthmore.edu/alejandre/magic.square.html
- ✓ http://www.qnx.com/%7Eomsingla/magic_squares_delaloubere.htm
- ✓ http://www.pse.che.tohoku.ac.jp/%7Emsuzuki/MagicSquare.html
- ✓ http://www.cut-the-knot.com/game_st.html
- ✓ http://user.chol.com/%7Ebrainstm/MagicSquare.htm
- ✓ Joseph Madachy, *Mathematics on Vacation*.
- ✓ J.A.H. Hunter & Joseph Madachy, *Mathematical Diversions*.