CHAPTER 4

Algorithm Implementation

To confirm the result we prove in chapter 3 that there are 384 pandiagonal Lanna Magic Squares, we also used mathematical program to find the result. The mathematical program we used is Scilab.

As the Lanna Magic Square is new to study in mathematics aspects, we want to discover not only finding all pandiagonal Lanna Magic Squares, but also finding its pandiagonal and semi pandiagonal of the Lanna Magic Squares.

Thereby, our procedures are first finding all Lanna Magic Squares. After that, finding all pandiagonal Lanna Magic Squares and last, findind all semi pandiagonal Lanna Magic Squares.

Let the numbers in each position be variables $a, b, c, \ldots, p$ as in the table 4.1 below

Table 4.1: A Lanna Magic Square

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

So, the square will be a Lanna Magic Square if and only if

$$a + b + c + d = 32 \tag{4.1}$$

$$e + f + g + h = 32 \tag{4.2}$$

$$i + j + k + l = 32 \tag{4.3}$$

$$m + n + o + p = 32 \tag{4.4}$$

$$a + e + i + m = 32 \tag{4.5}$$

$$b + f + j + n = 32 \tag{4.6}$$

$$c + g + k + o = 32 \tag{4.7}$$

$$d + h + l + p = 32 \tag{4.8}$$

$$a + f + k + p = 32 \tag{4.9}$$

$$d + g + j + m = 32 \tag{4.10}$$

First of all we try to reduce variables since if we make the algorithm for Equations 4.1 - 4.10, it would compute the permutation of $a, b, c, \ldots, p$ which are from the numbers $\{1, 2, 3, 4, 5, 6, 7, 8, 8, 9, 10, 11, 12, 13, 14, 15\}$. It has to compute $\frac{16!}{2!}$ which is computationally expensive and take too much time. Therefore, we reformulate equations 4.1 - 4.10. According to equations 4.5 - 4.8, we obtain

$$m = 32 - a - e - i \tag{4.11}$$

$$n = 32 - b - f - j \tag{4.12}$$

$$o = 32 - c - g - k \tag{4.13}$$

$$p = 32 - d - h - l \tag{4.14}$$

According to equations 4.11 - 4.14, we get

$$m + n + o + p = 4 \times 32(a + b + \ldots + l) \tag{4.15}$$

Substitute equations 4.1 - 4.3 into equation 4.15, we obtain

$$m + n + o + p = 32$$

This means that equation 4.4 is redundant. Then, substitute equation 4.14 into equation 4.9 and equation 4.15 into equation 4.10, we acquire

$$l = a + f + k - d - h \tag{4.16}$$

$$j = a + e + i - d - g \tag{4.17}$$

Substitute equations 4.16 - 4.17 into equation 4.3, we obtain

$$k = \frac{1}{2}(2d + g + h - 2a - e - f - 2i + 32) \tag{4.18}$$

Moreover, we from equation 4.1 and 4.2, we derive

$$d = 32 - a - b - c \tag{4.19}$$

$$h = 32 - e - f - g \tag{4.20}$$

Thus, variables $\{d,\ h,\ j,\ k,\ l,\ m,\ n,\ o,\ p\}$ can be determined by free variables $\{a,\ b,\ c,\ e,\ f,\ g,\ i\}$. Hence, to find all Lanna Magic Squares is to find all possible permutations of $\{a,\ b,\ c,\ e,\ f,\ g,\ i\}$ from $\{1, 2, 3, 4, 5, 6, 7, 8, 8, 9, 10, 11, 12, 13, 14, 15\}$ which is $\frac{\binom{16}{7}}{2!}$.

Here are each step used to compute all Lanna Magic Squares, pandiagonal Lanna Magic Squares and semi pandiagonal Lanna Magic Squares by program Scilab.

Step 1 : we set a matrix 1×16, $A = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15]$ and we create the function *combi* to list all possible combinatorics of $\{a, b, c, e, f, g, i\}$ from $\{1, 2, 3, 4, 5, 6, 7, 8, 8, 9, 10, 11, 12, 13, 14, 15\}$.

```
function p=combi(A)
    s=size(A)
    column=s(1,2)
    B=[ ]
        for a=1:column-6
          for b = a + 1:column-5
            for c = b+1:column-4
              for e = c + 1:column-3
                for f = e + 1:column-2
                  for g = f + 1:column-1
                    for i = g+1:column
            B=[ B ; [A(a) A(b) A(c) A(e) A(f) A(g) A(i)] ]
                    end
                  end
                end
              end
            end
          end
        end
    p=B
endfunction
```

After input matrix A into the function *combi*, we derive 11,440 results which some results maybe alike. So, we use the function *unique* to find the unique results and let it be a matrix B.

```
B = unique(A,:)
```

we get that there are 8,512 results of all unique possible combinatorics of $\{a, b, c, e, f, g, i\}$ from $\{1, 2, 3, 4, 5, 6, 7, 8, 8, 9, 10, 11, 12, 13, 14, 15\}$.

Step 2 : then we find all permutations of each result. It means that each result has to multiply 7! to get all permutations. We find all permutations by making the function *permrow* as below.

```
function g=permrow(A)
    s=size(A)
    row=s(1,1)
    M=[ ]
        for i=1:row
          M=[M;perms(A(i,:))]
        end
    g=M
endfunction
```

In this step, we get 42,497,280 permutations.

Step 3 : now we have all 42,497,280 possible numbers for $\{a, b, c, e, f, g, i\}$. We create the function *lanna* to find all Lanna Magic Squares as below.

```
function m=lanna(M)
  s=size(M)
  row=s(1,1)
  A=[ ]
  for i=1:row
  a=M(i,1), b=M(i,2), c=M(i,3), e=M(i,4), f=M(i,5), g=M(i,6), i=M(i,7)
    d = 32 - a - b - c
      if d > 0 & d < 16 then h = 32 - e - f - g
        if h > 0 & h < 16 then j = a + e + i - d - g
          if j > 0 & j < 16 then k = (2d + g + h - 2a - e - f - 2i + 32)/2
            if k > 0 & k < 16 then l = a + f + k - d - h
              if l > 0 & l < 16 then m = 32 - a - e - i
                if m > 0 & m < 16 then n = 32 - b - f - j
                  if n > 0 & n < 16 then o = 32 - c - g - k
                    if o > 0 & o < 16 then p = 32 - d - h - l
```

```
                    if p > 0 & p < 16 then
                      MS = [a b c d e f g h i j k l m n o p]
                      UNI=unique(MS)
                      if sum(UNI>0)==15 & sum(UNI<16)==15 then
                        A=[A;MS]
                      end
                    end
                end
              end
            end
          end
        end
      end
    end
  end
  end
  m=A
endfunction
```

The command *sum(UNI>0)==15* means to check if in the matrix UNI has the numbers more than zero are 15 numbers. Inthe same way, *sum(UNI<16)==15* to check the numbers less than 16 are 15 numbers. If the 2 conditions are true then we can guarantee that the answer matrix consists of the right number {1, 2, 3, 4, 5, 6, 7, 8, 8, 9, 10, 11, 12, 13, 14, 15}.

The results we get from this step are 10,080 squares which some of them maybe the same. So, we now take the function *unique* again and we derive all unique Lanna Magic Squares which have 8,512 squares in total.

Next, we find pandiagonal Lanna Magic Squares by creating the function *ispanlanna* in the function *panlannacheck* below.

```
function a=ispanlanna(B)
    if (B(1)+B(8)+B(11)+B(14)==32 & B(2)+B(5)+B(12)+B(15)==32 &
    B(3)+B(6)+B(9)+B(16)==32 & B(4)+B(5)+B(10)+B(15)==32 &
    B(3)+B(8)+B(9)+B(14)==32 & B(2)+B(7)+B(12)+B(13)==32)
        then a=%T
```

```
        else a=%F
    end
endfunction
```

```
function a=panlannacheck(B)
    s=size(B)
    row=s(1,1)
    C=[ ]
        for i=1:row
            if (ispanlanna(B(i,:))) then C = [C ; B(i,:)]
            end
        end
    a=C
endfunction
```

The result we obtained is 384. So, this can be confirmed from proving in chapter 3 that there are 384 pandiagonal Lanna Magic Squares.

And the last one, we find all semi pandiagonal Lanna Magic Squares by using the function *islannasemipan* in the function *semipanlannacheck* shown below.

```
function a=islannasemipan(B)
    if ( B(2)+B(5)+B(12)+B(15)==32 & B(3)+B(8)+B(9)+B(14)==32
    &B(1)+B(8)+B(11)+B(14)==32 & B(3)+B(6)+B(9)+B(16)==32 &
    B(4)+B(5)+B(10)+B(15)==32 & B(2)+B(7)+B(12)+B(13)==32)
        then a=%T
        else a=%F
    end
endfunction
```

```
function a=semipanlannacheck(B)
    s=size(B)
    row=s(1,1)
    C=[ ]
        for i=1:row
```

```
            if (islannasemipan(B(i,:)))  then
            C = [C ; B(i,:)]
            end
        end
    a=C
endfunction
```

We get that there are 3,072 semi pandiagonal Lanna Magic Squares.