数据结构实验报告

1. 实验序号与名称

实验 5. 2：螺旋方阵

2. 学号，姓名，专业，实验时间

学号：2016141223037　　姓名：宋运翔　　专业：计算金融（方向）　　实验时间：第 8~11 周。

3. 实验内容与目标

下面是一个 5*5 阶螺旋方阵，设计相关算法输出此形式的 n*n（n<20）阶阵（逆时针方向旋转）。

$$\begin{bmatrix} 1 & 16 & 15 & 14 & 13 \\ 2 & 17 & 24 & 23 & 12 \\ 3 & 18 & 25 & 22 & 11 \\ 4 & 19 & 20 & 21 & 10 \\ 5 & 6 & 7 & 8 & 9 \end{bmatrix}$$

4．实验工具

Microsoft Visual Studio 2013

5.实验分析

5. 1 基础算法分析

对于如下方阵 A, 可以看成是三个正方形组成，对于左上角下标为(i,i)，边长为 side 的正方形的其他三个顶点的下标依次为 $(i+side-1,i)$，$(i+side-1,i+side-1)$，$(i,i+side-1)$。

$$A=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}$$

因此填充该螺旋矩阵时，可以按照从外到内依次填充各正方形。

5. 2 主函数流程图

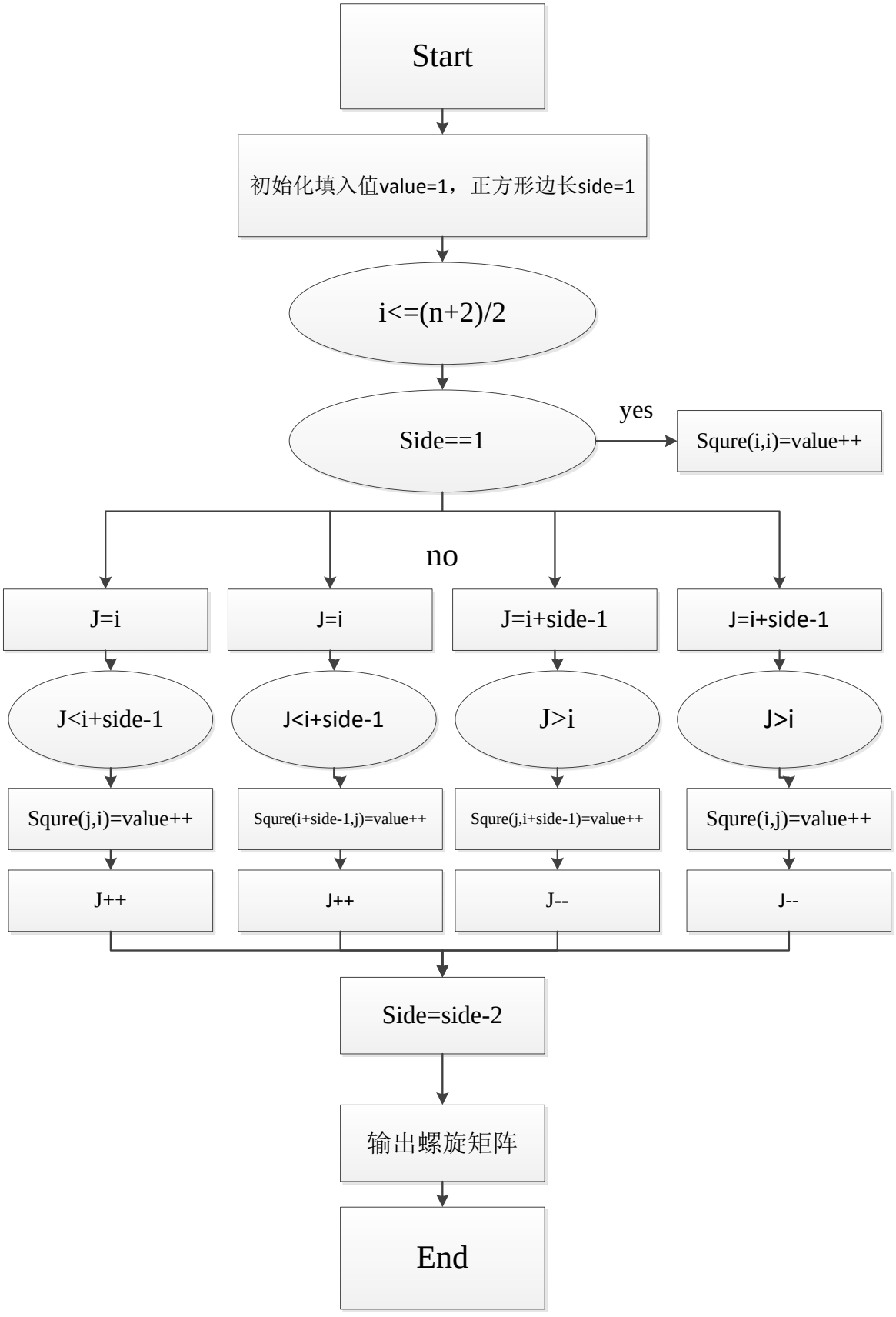

Start
初始化填入值value=1，正方形边长side=1
i<=(n+2)/2
Side==1
yes
Squre(i,i)=value++
no
J=i
J=i
J=i+side-1
J=i+side-1
J<i+side-1
J<i+side-1
J>i
J>i
Squre(j,i)=value++
Squre(i+side-1,j)=value++
Squre(j,i+side-1)=value++
Squre(i,j)=value++
J++
J++
J--
J--
Side=side-2
输出螺旋矩阵
End

图 5.1 主函数流程图

5. 3 主要算法代码实现

```
    //生成 n*n 阶的螺旋矩阵
    for (i = 1; i <= (n + 1) / 2; i++)
    {
        //生成第 i 个正方形，则其四个顶点的下标为（i,i）（i+side-1,i+side-1）（i+side-1,i）（i,i+side-1）
        if (side == 1)
        {
            squre(1, 1) = value++;
        }
        else
        {
            for (j = i; j < i + side - 1; j++)
            {
                //填入正方形的左边的边
                squre(j, i) = value;
                value++;
            }
            for (j = i; j < i + side - 1; j++)
            {
                //填入正方形的下边的边
                squre(i + side - 1, j) = value++;
            }
            for (j = i + side - 1; j > i; j--)
            {
                //填入正方形的右边的边
                squre(j, i + side - 1) = value++;
            }
            for (j = i + side - 1; j > i; j--)
            {
                //填入正方形的上边的边
                squre(i, j) = value++;
            }
        }
        side = side - 2;
    }
```

6. 实验步骤

1.建立工程 screw_squre_matryx。

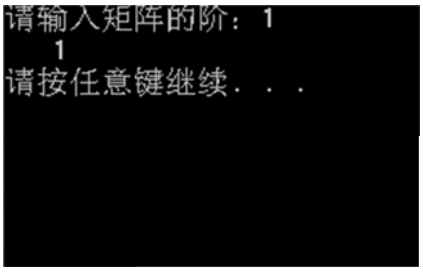
2.将软件包中的 utility.h 复制到 screw_squre_matryx 文件夹中，并将 utility.h 加入到工程中。

3.将矩阵需要的头文件 matrix.h 复制到 screw_squre_matryx 文件夹中，并将 matrix.h 加入到工程中。

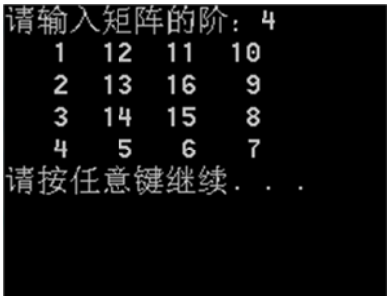
4.建立源程序文件 main.cpp，实现 main（）函数。

7. 测试与结论

7.1 输出 1 阶矩阵



7.2 输出 4 阶矩阵



7.3 输出 12 阶矩阵

从以上的显示可以看出，本程序满足实验目标要求。

8. 思考与感悟

该程序只有 main（）函数，同样，可以将该函数封装成一个函数，同样还可以定义螺旋方阵类，对该函数进行封装。

展开联想，该算法仅仅针对逆时针的螺旋方阵，可以提供两种选择，根据选择输出顺时针螺旋方阵或者逆时针螺旋方阵。该螺旋填充矩阵具有一定的趣味性，我不仅仅熟练了相关算法，还从中得到了快乐。