

# 人工智能技术实验报告——N皇后问题

姓名：杨帆 学号：1711503 专业：智能科学与技术

## 一、编译环境

- 1. 编程语言：  
C++
- 2. 可视化编写环境：  
QT

## 二、代码功能

### 1. 功能概述：

代码基于八皇后问题实现了N皇后问题的解法遍历，并将其可视化展现出来，其中包括动态的方案展示以及所有求解结果的展示，可随意设定皇后数量，最终求出结果。如下为10皇后问题的最终运行结果：



### 2. 核心代码（递归实现）：

```
1 void MainWindow::Queen(int row)
2 {
3     if (row == n) {
4         //此循环用于展示最终的所有解决方案
5         for (int i = 0; i < n; i++)
6             {
7                 QString a;
8                 a=QString::number(i+1,10);
```

```

9         QString b;
10         b=QString::number(chessboard[i]+1,10);
11         ui->textEdit_2->insertPlainText("(" +a+", "+b+")"+" ");
12     }
13     ui->textEdit_2->insertPlainText("\n");
14     SolveNum++; //解的数量
15     DisplaySolveNum();
16     init();
17     for(int i=0;i<n;i++)
18     {
19         ChessboardPaintArray[i][chessboard[i]]=3;
20     }
21     repaint();
22 }
23 else
24     for (int col = 0; col < n; col++)
25     {
26         chessboard[row] = col;
27         if (Judge(row)) //判断是否有棋子相吃的情况
28             Queen(row + 1); //没有棋子相吃的情况探索下一种情况
29     }
30 }

```

核心程序使用递归实现，首先输入皇后数 $n$ ，使用以为数组chessboard储存棋盘的情况，数组序号代表row，数组的值对应某一列的值，但判断函数返回false时，继续下一列再次判断，当最终循环 $n$ 次后，停止循环，展示最终结果。

有关界面部分的代码在程序中均有注释。

### 3.代码运行效率分析

N皇后问题的时间复杂度较高为 $O(n!)$ ，当 $N$ 较大时，会出现组合爆炸问题，难以解决。

### 三、实验总结

1. 通过此次实验更加深刻的了解并掌握了八皇后问题的原理，并且通过八皇后问题进一步拓展到N皇后问题的解决方案的实现；
2. 学会使用QT进行界面绘制以及exe程序的导出，发现QT的界面绘制与MFC相差不大，均比较复杂，后续算法的实现将考虑使用tkinter进行绘制。