项目说明文档

数据结构课程设计

——N皇后问题

作 者 姓 名： 陈君

学 号： 2250420

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji Universit

# 1 分析

## 1.1 背景分析

## N皇后问题（N-Queens Problem）是一个经典的组合优化问题，起源于国际象棋中的皇后棋子。问题的核心目标是在一个N×N的棋盘上放置N个皇后，使得它们在行、列和对角线方向上都不互相攻击。具体而言，任意两个皇后都不能在同一行、同一列或同一对角线上。这个问题的背景可以追溯到1848年，当时由德国数学家韦伯利（Max Bezzel）首次提出。它引起了数学家们的广泛关注，并成为计算机科学领域中经典的算法和人工智能问题之一。N皇后问题的复杂性主要在于其解空间的庞大性质。对于一个N×N的棋盘，解的数量随着N的增加呈指数级增长。因此，寻找问题的确切解对于传统的搜索算法来说是一项具有挑战性的任务。这个问题的解决方法通常涉及到深度优先搜索、回溯算法以及一些启发式搜索方法。其中，回溯算法是最常见的解决手段之一。该算法通过逐行放置皇后，并在每一步检查是否满足不互相攻击的条件，如果不满足则回溯到上一步重新选择位置。这一过程一直持续到所有皇后都被成功放置或者所有可能的位置都被尝试过。N皇后问题不仅仅是一个抽象的数学问题，它在实际中也有许多应用。类似的约束和优化问题出现在排产计划、电路设计、航空航天、网络布局等领域。通过解决N皇后问题，人们能够深入研究和理解组合优化问题的本质，并发展出更广泛领域的解决方案。

## 1.2 功能分析

八皇后在棋盘上分布的各种可能的格局数目非常大，约等于2的32次方种，但是，可以将一些明显不满足问题要求的格局排除掉。由于任意两个皇后不能同行，即每行只能放置一个皇后，因此将第i个皇后放在第i航上，这样在放置第i个皇后时，只要考虑它与前i-1个皇后处于不同列和不同对角线位置上即可。

解决这个问题采用回溯法，首先将第一个皇后放置在第一行第一列，然后，依次在下一行上放置一个皇后，直到八个皇后全部放置安全。在放置每个皇后时，都依次兑每一列进行检测，首先检测放在第一列是否与已放置的皇后冲突，如不冲突，则将皇后放置在该列，否则，选择改行的下一列进行检测。如整行的八列都冲突，则回到上一行，重新选择位置，依次类推。

# 2 设计

## 2.1 数据结构设计

二维字符数组 char queen[8][8]：用途：表示棋盘上的皇后放置情况。

分析：每个元素可以是 '0' 或 'X'，其中 '0' 表示该位置没有皇后，'X' 表示该位置放置了皇后。

布尔数组 bool bcolumn[8]：用途：表示每列是否有皇后。

分析：bcolumn[i] 的值为 true 表示第 i 列有皇后，为 false 表示没有皇后。

布尔数组 bool bMain\_diagonal[15] 和 bool bSub\_diagonal[15]：用途：表示主对角线和副对角线是否有皇后。

分析：bMain\_diagonal[i] 的值为 true 表示主对角线 i 有皇后，bSub\_diagonal[i] 类似。

## 2.4 系统设计

主程序首先说明游戏的规则：现有N×N的棋盘，放入N个皇后，要求所有皇后不在同一行，列和同一斜线上!然后提示用户输入皇后的个数。接着调用递归函数算出皇后可能摆法。

# 3 实现

### 3.1void place(int row,int num,char queen[8][8],bool bcolumn[8],

### bool bMain\_diagonal[15],bool bSub\_diagonal[15])；

具体实现：基于回溯算法，通过递归地尝试在每一行的每一列放置皇后，然后检查是否满足条件。如果某个位置可以放置皇后，就继续递归调用 place 函数放置下一行的皇后。如果某个位置无法放置皇后，那么通过回退操作，取消当前位置的皇后放置，继续尝试下一个位置。

递归终止条件：当 row 等于 num 时，说明已经成功摆放了 N 个皇后，此时调用 putout 函数输出当前的解，并返回。

1、遍历每一列：

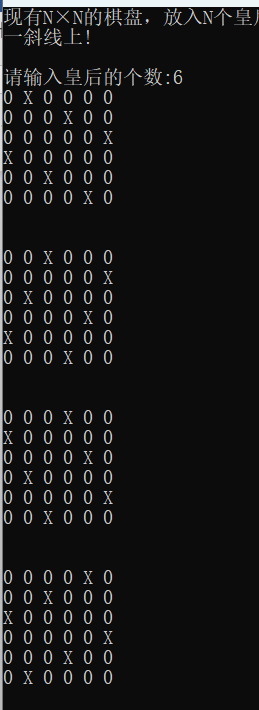
使用 for 循环遍历当前行的每一列，从第 0 列到 num - 1 列。如果这一列这一行的位置可以放置皇后就继续递归调用place函数。

2、在递归调用返回后，回退当前行的皇后放置，将对应位置的皇后标记为未放置状态，并同时更新辅助数组。

## 3.2void putout(char queen[8][8],int num);

具体实现：使用两个for循环输出生成的数组。

## 3.2void putout(char queen[8][8],int num);功能截图展示



## 3.6 总体系统的实现

主程序首先说明游戏的规则：现有N×N的棋盘，放入N个皇后，要求所有皇后不在同一行，列和同一斜线上!然后提示用户输入皇后的个数。接着调用递归函数算出皇后可能摆法。

### }3.6.1 总体系统截屏示例

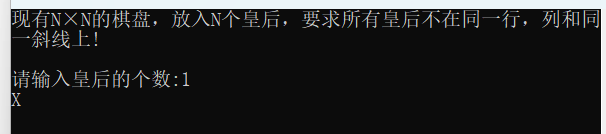


# 4 测试

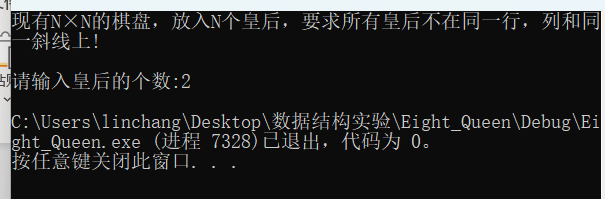
## 4.1 系统总体功能测试功能测试

不同皇后数量的测试：

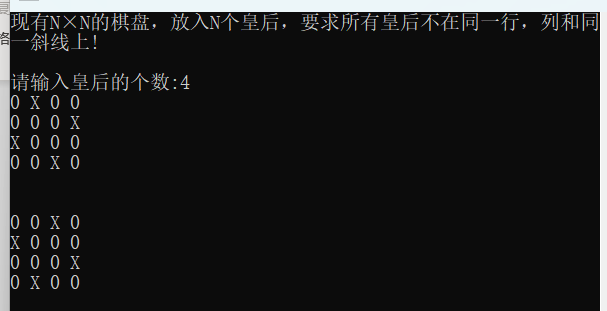
N=1：



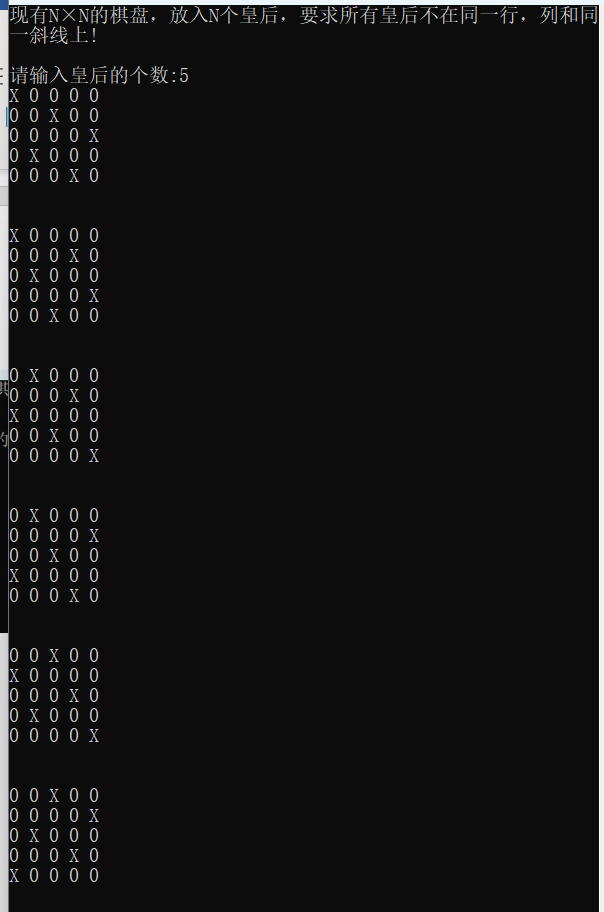
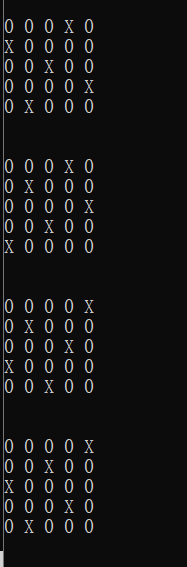
N=2：



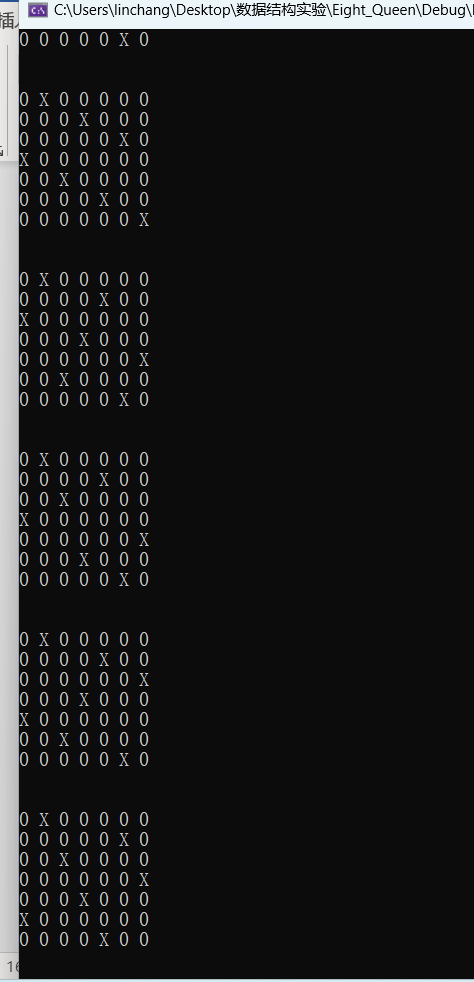
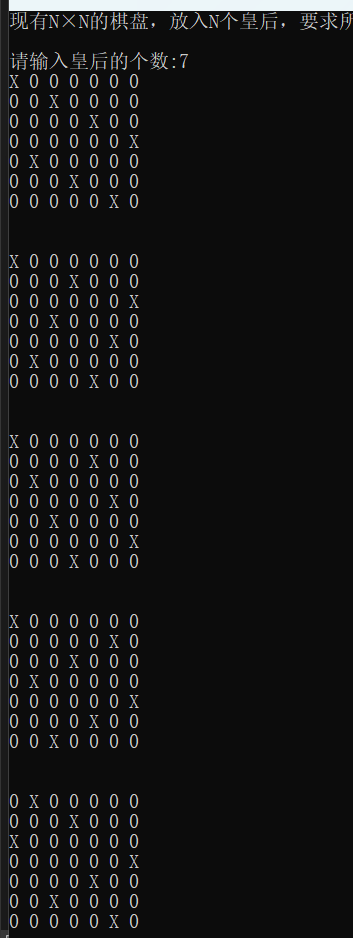
N=4：

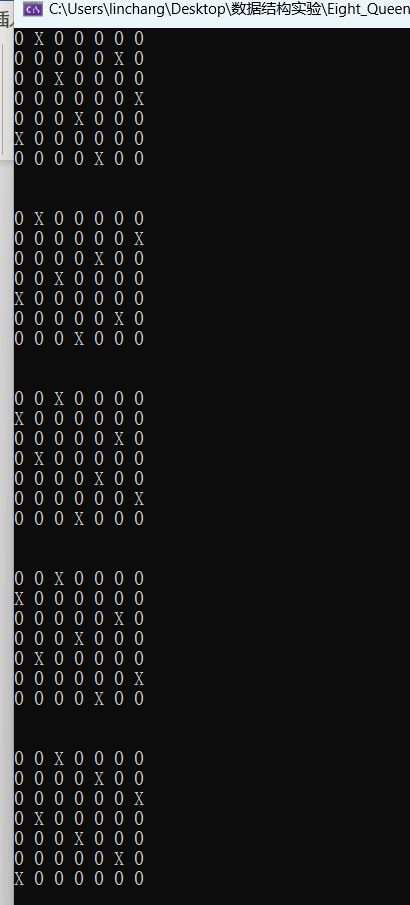
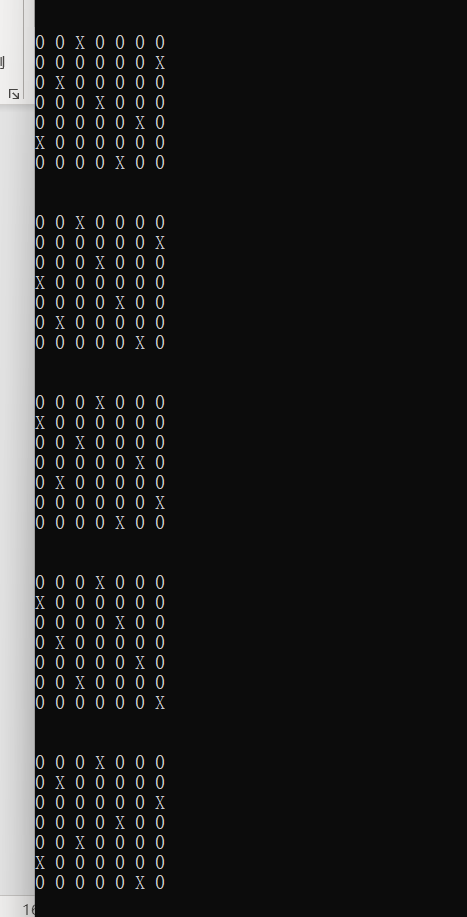


N=5：



N=7:





由于数量过多

N=8就截图展示了