**《N 皇后问题实验报告》**

**编号：2025-01**  
**姓名：刘鑫**  
**学号：2023141461105**  
**日期：2025年6月18日**

**一、问题描述**

N 皇后问题是一个经典的组合优化问题，要求在 N×N 的国际象棋棋盘上放置 N 个皇后，使得任意两个皇后不在同一行、同一列或同一条对角线上。该问题的求解属于典型的回溯算法应用场景。

**二、算法设计**

**2.1 基本思想**

使用 **回溯法（Backtracking）** 遍历搜索每一行的合法皇后位置。在搜索过程中，逐步构建解，并在不满足条件时进行回退（回溯），避免无效搜索。

**2.2 主要步骤**

1. 从第 0 行开始，尝试将皇后放入每一列；
2. 检查当前位置是否与之前的皇后冲突（列、主对角线、副对角线）；
3. 若不冲突，则递归放置下一行皇后；
4. 若已放置到最后一行，则记录当前解；
5. 回溯并尝试下一个可能位置。

**2.3 核心伪代码**

function backtrack(row):

if row == N:

记录当前解

return

for col in 0..N-1:

if 当前列和对角线合法:

放置皇后

backtrack(row + 1)

撤销放置（回溯）

**三、优化思路**

**3.1 剪枝策略**

为了提高效率，在放置每个皇后时，通过以下结构进行剪枝：

* 使用 set 存储已使用的 **列**；
* 使用 set 存储已使用的 **主对角线（row - col）**；
* 使用 set 存储已使用的 **副对角线（row + col）**。

通过这些剪枝，大大减少了不必要的递归和无效尝试。

**3.2 可扩展优化（未实现）**

* **对称性优化**：利用解的对称性减少重复计算；
* **位运算优化**：用二进制快速判断列与对角线是否被占用，进一步提高效率（适用于更大 N）；
* **并行搜索**：使用多线程处理不同列的分支搜索（适用于高性能环境）。

**四、实验结果与分析**

**4.1 实验平台**

* 开发语言：Python 3.11
* 编辑器：PyCharm Community Edition 2025.1.2
* 系统环境：Windows 11
* 测试函数：solve\_n\_queens(n)
* 计时方法：time.time() 记录函数执行前后时间

4.2 运行时间统计图（N = 4 到 12）

图表, 折线图

AI 生成的内容可能不正确。