

# N皇后问题报告

## 一、算法说明

本实验采用回溯法解决N皇后问题：

- 1.逐行放置：从第一行开始，依次在每行放置一个皇后
- 2.冲突检测：使用is\_safe()函数检查当前位置是否与已放置皇后冲突（同列/同对角线）
- 3.回溯机制：当某行无法放置皇后时，回退到上一行调整位置
- 4.终止条件：成功放置N个皇后时记录解，或回溯到第一行仍无解时结束

## 二、优化思路

- 1.剪枝策略：提前排除不安全位置，减少无效搜索
- 2.对称性优化：利用棋盘对称性减少重复计算（未实现）
- 3.按需求解：支持只找第一个解，大幅减少大N时的计算量
- 4.输出控制：N>8时不打印全部解，避免输出过载

## 三、实验结果

### ✓ 运行成功

```
请输入棋盘大小N (N≥4): 4
输出所有解? (y/n): y
是否运行实验分析? (y/n): y
找到 2 个解:
解 1:
. Q . .
. . . Q
Q . . .
. . Q .

解 2:
. . Q .
Q . . .
. . . Q
. Q . .

计算耗时: 0.0000秒
N=4: 0.0000秒
N=5: 0.0001秒
N=6: 0.0005秒
N=7: 0.0017秒
N=8: 0.0080秒
N=9: 0.0400秒
N=10: 0.2078秒
N=11: 1.1711秒
N=12: 6.9744秒
```

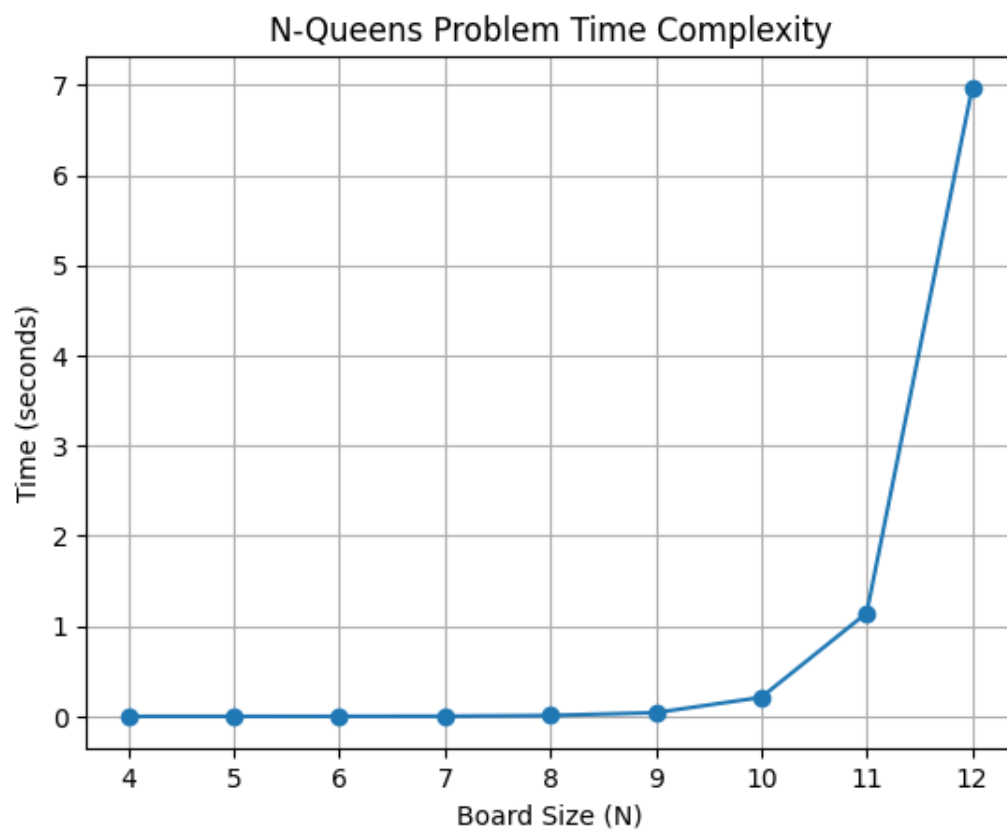
### ✓ 运行成功

```
请输入棋盘大小N (N≥4): 8
输出所有解? (y/n): y
是否运行实验分析? (y/n): y
找到 92 个解:
解 1:
Q . . . . . . .
. . . . Q . . .
. . . . . . . Q
. . . . . Q . .
. . Q . . . . .
. . . . . . Q .
. Q . . . . . .
. . . Q . . . .

解 2:
Q . . . . . . .
. . . . . Q . .
. . . . . . . Q
. . Q . . . . .
. . . . . . Q .
. . . Q . . . .
. Q . . . . . .
. . . . Q . . .

解 3:
Q . . . . . . .
. . . . . . Q .
. . . Q . . . .
. . . . . Q . .
. . . . . . . Q
. Q . . . . . .
. . . . Q . . .
. . Q . . . . .

解 4:
Q . . . . . . .
. . . . . . Q .
. . . . Q . . .
. . . . . . . Q
. Q . . . . . .
. . . Q . . . .
. . . . . Q . .
. . . . . . . Q
```



#### 四、改进方向

位运算优化列和对角线冲突检测

多线程并行搜索不同分支

基于对称性的解去重