

N 皇后问题求解报告

一、算法说明

1. 问题定义

N 皇后问题是在一个 $N \times N$ 的棋盘上放置 N 个皇后，使得任意两个皇后不能在同一行、同一列或同一对角线上。这是一个典型的回溯算法问题。

2. 解决方案

我们采用回溯法来求解 N 皇后问题。基本思想是：逐行放置皇后，每次在当前行尝试所有可能的位置，如果找到一个合法位置（不与已放置的皇后冲突），则继续递归地放置下一行的皇后；如果当前行所有位置都不合法，则回溯到上一行，改变上一行皇后的位置，继续搜索。

3. 冲突检测

为了检测新放置的皇后是否与已放置的皇后冲突，我们设计了冲突检测函数。该函数检查新皇后是否与已有皇后在同一列或同一对角线上。我们使用三个列表来分别记录已放置皇后的列位置、主对角线位置和副对角线位置，从而实现 $O(1)$ 时间复杂度的冲突检测。

二、对称性剪枝优化思路 3.1.1 对称性原理

1. 棋盘具有多种对称性，包括旋转对称性和反射对称性。

具体来说，一个 $N \times N$ 的棋盘可以通过以下对称变换生成相同的解：

旋转对称性：

旋转 90 度

旋转 180 度

旋转 270 度

反射对称性：

水平反射（上下翻转）

垂直反射（左右翻转）

对角线反射（主对角线翻转）

反对角线反射（副对角线翻转）

这些对称变换意味着，如果我们找到一个解，可以通过这些变换生成其他对称的解。因此，我们只需要找到其中一个解，然后通过对称变换生成其他解，从而减少需要搜索的解空间。

2. 具体实现

为了利用对称性剪枝，我们可以在搜索过程中只考虑一部分布局，然后通过对称变换生成其余的解。具体实现步骤如下：

限制第一列的皇后位置：

在第一列中，我们只考虑将皇后放置在第一行到 $N/2$ 行。这是因为放置在 $N/2$ 行以上的位置会通过对称变换生成与放置在 $N/2$ 行以下位置相同的解。

例如，对于 $N=8$ 的棋盘，我们只考虑将皇后放置在第一列的第 1 行到第 4 行。

生成对称解：

对于每一个找到的解，通过上述对称变换生成其他对称的解。

例如，对于每一个解，我们可以生成其旋转 90 度、180 度、270 度的解，以及水平反射、垂直反射、对角线反射和反对角线反射的解。

三，测试截图：

```
Enter the number of queens (N >=4): 4
Calculating solutions...
Found 4 solutions for 4-Queens problem.
Time taken: 0.0000 seconds
Print all solutions? (y/n): y
. Q . .
. . . Q
Q . . .
. . Q .

. . Q .
Q . . .
. . . Q
. Q . .

. . . Q
. . Q .
Q . . .
Q . . .

Q . . .
. Q . .
. . Q .
. . . Q
```

```
Enter the number of queens (N >=4): 8
Calculating solutions...
Found 94 solutions for 8-Queens problem.
Time taken: 0.0337 seconds
Print all solutions? (y/n): y
Q . . . . . . .
. . . . Q . . .
. . . . . Q . .
. . . . . . Q .
. . . . . . . Q
. . . . . Q . .
. . . . . . . Q
. . . . . . . Q
Q . . . . . . .
. . . . . Q . .
. . . . . . Q .
. . . . . . . Q
. . . . . Q . .
. . . . . . . Q
. . . . . . . Q
. . . . . Q . .
Q . . . . . . .
```