数据结构

第二次上机作业实验文档

学号2017211123

班级 2017211301

序号 17

姓名 褚逸豪

## **问题描述**

如何能够在 8×8 的国际象棋棋盘上放置八个皇后，使得任何一个皇后都无法直接吃掉其他的皇后？

## 算法思路

递归枚举每行的摆放方案，用三个数组分别标识标识每一纵列，每一斜行，每一反斜行是否已被放上棋子，利用这些数组来进行条件约束。循环枚举某行摆放的位置，递归枚举下一行。每次完整的枚举完八行后计数并输出方案。在回溯时退掉标记。

## 算法描述

1. 初始化数组v（列），x（\），y（/）为0值
2. 递归（参数1 位置）：
   1. 如果位置为8，打印合法序列，并进行计数
   2. 枚举当前行的每一个位置
   3. 如果对应的v，x，y均为0值，则将对应的v，x，y设为1，序列的当前位置设为当前枚举的行中位置，并递归下一个位置。
   4. 回溯时退掉当前位置对应的v，x，y标记

## 源程序及驱动程序

编译命令：g++ <文件名.cpp>

#include <cstdio>

#include <iostream>

#include <cstring>

using namespace std;

namespace eight\_queens {

const int MXN = 31;

bool x[MXN], y[MXN], v[MXN];

int seq[MXN];

int cnt;

void dfs(int pos) {

if (pos == 8) { // 得到一组合法序列，输出并计数

for (int i = 1; i <= 8; ++i) {

printf("%d ", seq[i]);

}

++cnt;

putchar('\n');

return;

}

for (int i = 1; i <= 8; ++i) {

if (!v[i] && !x[i - pos + 8] && !y[i + pos]) { // 当前位置纵列、反斜行、斜行是否已经有元素

v[i] = 1, x[i - pos + 8] = 1, y[i + pos] = 1; // 打上标记

seq[pos + 1] = i;

dfs(pos + 1); // 下个位置

v[i] = 0, x[i - pos + 8] = 0, y[i + pos] = 0; // 回退标记

}

}

}

void solve() {

cnt = 0;

memset(seq, 0, sizeof(seq)); // 初始化各种数组

memset(x, 0, sizeof(x));

memset(y, 0, sizeof(y));

memset(v, 0, sizeof(v));

dfs(0);

printf("%d\n", cnt);

}

};

int main() {

eight\_queens::solve();

}

## 测试数据

### Input

无

### Output

1 5 8 6 3 7 2 4

1 6 8 3 7 4 2 5

1 7 4 6 8 2 5 3

1 7 5 8 2 4 6 3

2 4 6 8 3 1 7 5

2 5 7 1 3 8 6 4

2 5 7 4 1 8 6 3

2 6 1 7 4 8 3 5

2 6 8 3 1 4 7 5

2 7 3 6 8 5 1 4

2 7 5 8 1 4 6 3

2 8 6 1 3 5 7 4

3 1 7 5 8 2 4 6

3 5 2 8 1 7 4 6

3 5 2 8 6 4 7 1

3 5 7 1 4 2 8 6

3 5 8 4 1 7 2 6

3 6 2 5 8 1 7 4

3 6 2 7 1 4 8 5

3 6 2 7 5 1 8 4

3 6 4 1 8 5 7 2

3 6 4 2 8 5 7 1

3 6 8 1 4 7 5 2

3 6 8 1 5 7 2 4

3 6 8 2 4 1 7 5

3 7 2 8 5 1 4 6

3 7 2 8 6 4 1 5

3 8 4 7 1 6 2 5

4 1 5 8 2 7 3 6

4 1 5 8 6 3 7 2

4 2 5 8 6 1 3 7

4 2 7 3 6 8 1 5

4 2 7 3 6 8 5 1

4 2 7 5 1 8 6 3

4 2 8 5 7 1 3 6

4 2 8 6 1 3 5 7

4 6 1 5 2 8 3 7

4 6 8 2 7 1 3 5

4 6 8 3 1 7 5 2

4 7 1 8 5 2 6 3

4 7 3 8 2 5 1 6

4 7 5 2 6 1 3 8

4 7 5 3 1 6 8 2

4 8 1 3 6 2 7 5

4 8 1 5 7 2 6 3

4 8 5 3 1 7 2 6

5 1 4 6 8 2 7 3

5 1 8 4 2 7 3 6

5 1 8 6 3 7 2 4

5 2 4 6 8 3 1 7

5 2 4 7 3 8 6 1

5 2 6 1 7 4 8 3

5 2 8 1 4 7 3 6

5 3 1 6 8 2 4 7

5 3 1 7 2 8 6 4

5 3 8 4 7 1 6 2

5 7 1 3 8 6 4 2

5 7 1 4 2 8 6 3

5 7 2 4 8 1 3 6

5 7 2 6 3 1 4 8

5 7 2 6 3 1 8 4

5 7 4 1 3 8 6 2

5 8 4 1 3 6 2 7

5 8 4 1 7 2 6 3

6 1 5 2 8 3 7 4

6 2 7 1 3 5 8 4

6 2 7 1 4 8 5 3

6 3 1 7 5 8 2 4

6 3 1 8 4 2 7 5

6 3 1 8 5 2 4 7

6 3 5 7 1 4 2 8

6 3 5 8 1 4 2 7

6 3 7 2 4 8 1 5

6 3 7 2 8 5 1 4

6 3 7 4 1 8 2 5

6 4 1 5 8 2 7 3

6 4 2 8 5 7 1 3

6 4 7 1 3 5 2 8

6 4 7 1 8 2 5 3

6 8 2 4 1 7 5 3

7 1 3 8 6 4 2 5

7 2 4 1 8 5 3 6

7 2 6 3 1 4 8 5

7 3 1 6 8 5 2 4

7 3 8 2 5 1 6 4

7 4 2 5 8 1 3 6

7 4 2 8 6 1 3 5

7 5 3 1 6 8 2 4

8 2 4 1 7 5 3 6

8 2 5 3 1 7 4 6

8 3 1 6 2 5 7 4

8 4 1 3 6 2 7 5

92

### Online Test

<https://ideone.com/WXWtMD>

## 结果分析

共92种方案

方案数与方案均正确

## 结论

使用递归加上正确的约束条件进行递归枚举，可以得到全部的正确结果

## 心得体会

暴力枚举是一项高中竞赛时十分熟练的技能