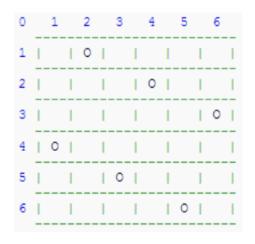
逻辑航线信息学奥赛系列教程

P1219 [USAC01.5]八皇后 Checker Challenge

题目描述

一个如下的 6×6 的跳棋棋盘,有六个棋子被放置在棋盘上,使得每行、每列有且只有一个,每条对角线(包括两条主对角线的所有平行线)上至多有一个棋子。



上面的布局可以用序列 2 4 6 1 3 5 来描述, 第 iii 个数字表示在第 i 行的相应位置有一个棋子,如下:

行号 1 2 3 4 5 6

列号 2 4 6 1 3 5

这只是棋子放置的一个解。请编一个程序找出所有棋子放置的解。 并把它们以上面的序列方法输出,解按字典顺序排列。 请输出前 3 个解。最后一行是解的总个数。

输入格式

一行一个正整数 nnn, 表示棋盘是 n×nn \times nn×n 大小的。

输出格式

前三行为前三个解,每个解的两个数字之间用一个空格隔开。第四行只有一个数字,表示解的总数。

输入样例

6

输出样例

解析

经典的八皇后深搜问题,注意地图的行列顺序,以及对角线的判断方法。 由于本题数据规模较大,因此我们使用了更为高效的标记结构来进行数据的存储。

编码

```
#include <bits/stdc++.h>
using namespace std;
//索引: 当前的行号, 值: 当前的列号
int box[15];
//记录前面行的这个位置是否有值
int check1[100];
//左对角线数组: 当前的行号+列号, 值: 0,1代表是否使用
int check2[100];
// 右对角线数组: 当前的行号+列号, 值: 0,1代表是否使用
int check3[100];
int col; //一共的行列数
int total; //总共的方案数
void print() {
   //遍历每一行
  for (int j = 1; j <= col; j++) {
      printf("%d ", box[j]);
   printf("\n");
}
//检测当前行是否可以放置
bool Check(int row, int col) {
   //判断是否在相同的列上,有则无法摆放
  bool colRes = check1[col];
   //判断主对角线是否存在皇后
  bool line1Res = check2[row + col];
   //判断副对角线是否存在皇后
  bool line2Res = check3[row - col + 15];
   //不满足条件
  if (colRes || line1Res || line2Res) {
      return false;
   }
```

```
return true;
}
void dfs(int row) {
   //已经全部找到,可以进行打印了,一共有step列
   if (row > col) {
       //记录找到了一种方案数
      total++;
       //最多打印三个
      if (total <= 3) {
          print();
       }
       return;
   }
   //总计8列
   for (int i = 1; i <= col; i++) {
       bool res = Check(row, i);
       //当前位置可以摆放
      if (res) {
           //将当前的位置进行摆放
         box[row] = i;
           check1[i] = 1;
           check2[row + i] = 1;
           check3[row - i + 15] = 1;
           //向下一个盒子进行放置
         dfs(row + 1);
           //取消占位
         check1[i] = 0;
           check2[row + i] = 0;
           check3[row - i + 15] = 0;
       }
   }
}
int main(int argc, char **argv) {
   cin >> col;
   dfs(1);
   printf("%d", total);
   return 0;
}
```

逻辑航线培优教育,信息学奥赛培训专家。

扫码添加作者获取更多内容。

