

C趣味编程百例(10)素数幻方 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/135/2021_2022_C_E8_B6_A3_E5_91_B3_E7_BC_96_c97_135586.htm 35.素数幻方 求四阶的素数幻方。即在一个4X4的矩阵中，每一个格填入一个数字，使每一行、每一列和两条对角线上的4个数字所组成的四位数，均为可逆素数。*问题分析与算法设计 有了前面的基础，本题应当说是不困难的。最简单的算法是：采用穷举法，设定4X4矩阵中每一个元素的值后，判断每一行、每一列和两条对角线上的4个数字组成的四位数是否都是可逆素数，若是则求出了满足题意的一个解。这种算法在原理是对的，也一定可以求出满足题意的全部解。但是，按照这一思路编出的程序效率很低，在微机上几个小时也不会运行结束。这一算法致命的缺陷是：要穷举和判断的情况过多。充分利用题目中的“每一个四位数都是可逆素数”这一条件，可以放弃对矩阵中每个元素进行的穷举的算法，先求出全部的四位可逆素数(204个)，以矩阵的行为单位，在四位可逆素数的范围内进行穷举，然后将穷举的四位整数分解为数字后，再进行列和对角线方向的条件判断，改进的算法与最初的算法相比，大大地减少了穷举的次数。考虑矩阵的第一行和最后一行数字，它们分别是列方向四位数的第一个数字和最后一个数字，由于这些四位数也必须是可逆素数，所以矩阵的每一行和最后一行中的各个数字都不能为偶数或5。这样穷举矩阵的第一行和最后一行时，它们的取值范围是：所有位的数字均不是偶数或5的四位可逆数。由于符合这一条件的四位可逆素数很少，所以这一范围限制又一次减少了穷举的次数。对算法

的进一步研究会发现：当设定了第一和第二行的值后，就已经可以判断出当前的这种组合是否一定是错误的(尚不能肯定该组合一定是正确的)。若按列方向上的四个两位数与四位可逆数的前两位矛盾(不是其中的一种组合)，则第一、二行的取值一定是错误的。同理在设定了前三行数据后，可以立刻判断出当前的这种组合是否一定是错误的，若判断出矛盾情况，则可以立刻设置新的一组数据。这样就可以避免将四个数据全部设定好以后再进行判断所造成的低效。根据以上分析，可以用伪语言描述以上改进的算法：开始找出全部四位的可逆素数；确定全部出现在第一和最后一行的四位可逆素数；在指定范围内穷举第一行在指定范围内穷举第二行若第一、第二、三行已出现矛盾，则继续穷举下一个数；在指定范围内穷举第四行判断列和对角方向是否符合题意若符合题意，则输出矩阵；否则继续穷举下一个数；结束在实际编程中，采用了很多程序设计技巧，假如设置若干辅助数组，其目的就是要最大限度的提高程序的执行效率，缩短运行时间。下面的程序运行效率是比较高的。*程序与程序注释

```
#include<stdio.h>
#include<string.h>
int number[210][5]. /*存放可逆素数及素数分解后的各位数字*/
int 0select[110]. /*可以放在矩阵第一行和最后一行的素数的下标*/
int array[4][5]. /*4X4的矩阵，每行0号元素存可逆素数对应的数组下标*/
int count. /*可逆素数的数目*/
int seletcount. /*可以放在矩阵第一行和最后一行的可逆素数的数目*/
int larray[2][200]. /*存放素数前二、三位数的临时数组所对应的数量计数器*/
int lcount[2].
int num(int number).
int ok(int number).
100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com
```