4皇后问题的解

数据结构：四个向量<Xi,F(Xi)>

思路：

递归实现

每一行检查前置的棋子是否与之互斥，若互斥，改变列数，若所有列数都不满足，结束递归，若成功递归至第4层，输出全部4个向量，即为所求解

若

X1=x2//行数相等

f（x1）=f（x2）//列数相等

|x1-x2|=|f（x1）-f（x2）|//同对角线

则不符合要求

4皇后问题源程序：

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

#include<conio.h>

#define N 4 //旗子数和棋盘大小

int a[N];

void show();

int check(int n);

void put(int n);

int main(){

system("color 2a");

put(0);

return 0;

}

void show(){ //输出一个正确结果

int i;

int p,q;

int b[4][4]={0};

static int t=1;

printf("第%d个解为：",t++);

for(i=0;i<N;i++){

b[i][a[i]]=1;

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

}

printf("\n");

for(p=0;p<4;p++){

for(q=0;q<4;q++){

if(b[p][q]==1)

printf("\*");

else

printf("0");

}

printf("\n");

}

}

int check(int n){ //检查当前棋子是否与前置旗子互斥

int i;

for(i=0;i<n;i++){

if (a[i]==a[n]||fabs(n-i) == fabs(a[i]-a[n]))

return 1;

}

return 0;

}

void put(int n){ //递归，每次检查与前面的条件是否矛盾

int i;

if(n == N)

return;

for(i = 0;i < N;i++){

a[n] = i;

if(! check(n)){

if(n == N-1)

show();

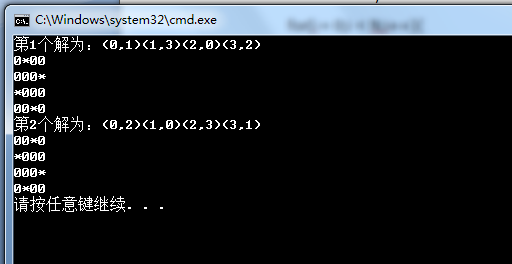
else

put(n+1);

}

}

}

实验结果：

实验总结：通过网上学习，了解了表示同一对角线的简单方法，对递归算法有了更熟练度 的应用