程序设计

1. 全局变量定义

//全局变量 记录解法个数

int sum;

2. 模块划分

(1)判断模块 判断皇后摆放是否符合要求 judge(int\* queen,int current\_column)

(2)输出模块 将每个解输出 print\_ans(int\* queen)

(3)搜索模块 运用深度搜索找出符合要求的所有解 find(int i,int\*queen,int n)

3.函数功能说明

// 参数：queen[n]下标代表行号，每一项的值代表列号

// current\_column表示当前列

// 返回0表示不符合要求返回1表示符合要求

// 判断当前皇后位置是否符合要求

(1) int judge(int\* queen,int current\_column){

for(int j=1;j<=current\_column;j++){

if((abs(queen[j]-queen[current\_column])==abs(j-current\_column) || queen[j]==queen[current\_column]))

return 0;

}

return 1;

}

// 遍历数组 在一行输出数组的值(末尾允许空格)

(2) void print\_ans(int\* queen,int n);

// 求解并输出解

(3) void find(int i,int\*queen,int n){

//递归出口，每出现一次，意味着得到了一种解

if(i>n){

sum++;

//输出一个解

print\_ans(queen);

return ;

}

//每一行8列都尝试走一遍

for(int j=1;j<=n;j++){

queen[i]=j;

// 判断当前皇后是否满足要求，满足则递归至下一层

if(judge(queen,i)){

find(i+1);

}

}

}

// 参数：n代表问题规模(n=8时即为8皇后问题)

// 初始化全局变量并运行find函数

(4) void solve(int n){

init();

};

// 初始化全局变量

(5) void init(){

sum=0;

}

(6) int main(void){

int n=0;

printf("请输入皇后个数:");

scanf("%d",&n);

solve(n);

printf("%d",sum);

return 0;

}