4皇后问题：

# include <stdio.h>

# define N 4 // 可以调整

int count=0; // 判断 Q[i][j] 是否存在冲突

int isCorrect(int i, int j, int (\*Q)[N]) {

int s,t; // 判断行

for(s=i,t=0; t<N; t++)

if(Q[s][t]==1 && t!=j)

return 0; // 判断列

for(s=0,t=j; s<N; s++)

if(Q[s][t]==1 && s!=i)

return 0; // 判断左上角

for(s=i-1,t=j-1; s>=0&&t>=0; s--,t--)

if(Q[s][t]==1) return 0; // 右下角

for(s=i+1,t=j+1; s<N&&t<N; s++,t++)

if(Q[s][t]==1) return 0; // 右上角

for(s=i-1,t=j+1; s>=0&&t<N; s--,t++)

if(Q[s][t]==1) return 0; // 左下角

for(s=i+1,t=j-1; s<N&&t>=0; s++,t--)

if(Q[s][t]==1)

return 0;

return 1; } // 递归函数，第 i+1 行的遍历

void Queue(int i, int (\*Q)[N]) {

int j,k; // 第四行已经遍历完，打印符合条件的结果，结束

if(i==N) {

count++;

printf("No.%d slove way\n",count);

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

for(j=0; j<N; j++)

printf("%-5d", Q[k][j]);

printf("\n"); //return;

}

printf("\n");

} // 遍历第 i+1 行的 j+1 列

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

if(isCorrect(i,j,Q)) {

Q[i][j]=1;

Queue(i+1,Q); // 遍历下一行

Q[i][j]=0; // 如果下一行不成立，则回溯，重置 0

}

}

}

int main() {

int Q[N][N];

int i,j; // 初始化二维数组并打印

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

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

Q[i][j] = 0;

printf("%-5d",Q[i][j]);

}

printf("\n");

} printf("\n");

Queue(0, Q); // 从第一行开始递归

printf("solve count is %d\n", count);

getchar();

getchar();

return 0;

}

****

Completeness：Yes

Optimality：Yes

Time complexly:最坏情况下o（n的n+1次幂）

Space complexly:o(n)