1.问题描述：

一个4×4国际象棋盘，依次放入四个皇后，条件：每行、每列及对角线上只允许出现一枚棋子。设：DATA=L（表）    x∈L    x ∈﹛i j﹜    1≤ i, j ≤4  
其中：i j 表示棋子所在行列

1. 代码：

**# include <stdio.h># define N 4**

**int** count=0;

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

**int** s,t;

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

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

**return** 0;

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

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

**return** 0;

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

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

**return** 0;

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

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

**return** 0;

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

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

**return** 0;

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

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

**return** 0;

**return** 1;

}

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

**int** j,k;

**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");

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;

}

3. 分析BFS

（1）完整性：如果有限且目标存在，BFS一定能找到：若目标不存在且为无限大，则BFS不会找到目标。

（2）最优性 ：改进BFS使其得到最优解：找到一个目标结点时，不是立即离开，而记录下目标结点的路径和值，如果有多个目标结点就加以比较，留下较优的结点，把所有可能的路径都搜索完之后，才输出记录的最优路径。

（3）时间复杂度：假设搜索树的状态空间中每个状态都有b个孩子，搜索树的根结点生成第一层的b个结点，每个子结点又生成b个子结点，于是第二层有了C:\Users\dell\AppData\Local\Temp\ksohtml\wpsDFAB.tmp.jpg个结点，以此类推，可以得到1+b+C:\Users\dell\AppData\Local\Temp\ksohtml\wpsDFAC.tmp.jpg+…..+C:\Users\dell\AppData\Local\Temp\ksohtml\wpsDFAD.tmp.jpg=O（C:\Users\dell\AppData\Local\Temp\ksohtml\wpsDFAE.tmp.jpg）个结点总数，其中d为深度，O（C:\Users\dell\AppData\Local\Temp\ksohtml\wpsDFBF.tmp.jpg）也就是BFS的时间复杂度。

（4）空间复杂度：最好的情况下，每层只包含一个结点，故空间复杂度为O（1），最坏情况下将会储存（C:\Users\dell\AppData\Local\Temp\ksohtml\wpsDFC0.tmp.jpg-1）个结点和一个无用的树，这时空间复杂度为O（C:\Users\dell\AppData\Local\Temp\ksohtml\wpsDFC1.tmp.jpg）。

4.分析：

在搜索时，对当前状态（搜索开始时，当前状态是初始状态）依次检测每一条规则，在当前状态未使用过的规则中找到第一条可应用规则，应用于当前状态，得到的新状态重新设置为当前状态，并重复以上搜索。如果当前状态无规则可用，或者所有规则已经被试探过仍未找到问题的解，则将当前状态的前一个状态（即直接生成该状态的状态）设置为当前状态。重复以上搜索，直到找到问题的解，或者试探了所有可能后仍找不到问题的解为止。