四皇后问题：

**状态空间**：

将每个可行解表示为一个4维向量<x1,x2,x3,x4>,其中xi表示第i行放置皇后的位置（列号），所有可行解为4维向量构成的集合{<x1,x2,x3,x4> | 1<=xi<=4,1<=i<=4},将它排列为完全四叉树。

四个皇后的所有可能摆放位置，既问题的解空间：16\*15\*14\*13=43680

若考虑行冲突，问题的解空间变为：16\*12\*8\*4=6144

若考虑列冲突，问题的解空间变为：16\*12\*8\*4=6144

若同时考虑行列冲突，问题的解空间变为：16\*9\*4\*1=576

实验代码：

**操作**：

X[i] = X[s]，则第i行和第s行皇后在同一列上

如果第i行的皇后在第j列，第s行的皇后在第t列，即X[i] = j 和 X[s] = t，则只要i - j = s- t 或者 i + j = s + t，说明两个皇后在对角线上，对两个等式进行变换后，得到结论，只要|i- s| = |j - t|（即|i- s| = |x[i] - x[s]|），则皇后在同一对角线上

#include<stdio.h>

#define N 4

char board[N][N];

int t;

int col[N];

int safetyPlace(int x,int y)

{

int i,j;

for(i=0;i<x;i++)

{

j=col[i];

if(x==i||y==j)

return 0;

if(x-y==i-j||x+y==i+j)

return 0;

}

return 1;

}

void get\_position(int i)

{

int w,j;

char a[1]={3};

if(i==N)

{

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

{

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

{

if(board[w][j]==001)

printf("%c ",board[w][j]);

else

{

printf("%c",a[0]);

printf("%c",board[w][j]);

}

}

printf("\n");

}

printf("\n");

printf("---------\n");

t++;

}

else

{

int u;

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

{

if(safetyPlace(i,u)==1)

{

col[i]=u;

board[i][u]=001;

get\_position(i+1);

col[i]=0;

board[i][u]=0;

}

}

}

}

main()

{

printf("%c是皇后!\n\n",001);

get\_position(0);

printf("一共有%d种方法! \n",t);

}