4皇后问题的解

**（1）状态空间**：

将该问题的每个可行解表示为一个4维向量<a1,a2,a3,a4>,其中a[i]表示第i行放置皇后的位置（列号）1<=a[i]<=4,1<=i<=4},排列为完全四叉树。

**（2）操作**：

皇后问题每个皇后不能在同行同列以及对角线上

if：

a[I，] =a[m，]，则第i行和第m行皇后在同一列上

else if：

a[,j]=a[,n]，则第j列和第n列皇后在同一行上

else if：

设a[I,j]为第i行，第j列皇后，a[m,n]为第m行，第n列皇后，

如果i-j=m-n或者i+j=s+t，为两个皇后在对角线上

**N皇后问题源程序：**

# 检测（x,y）这个位置是否合法（不会被其他皇后攻击到）

def is\_attack(queue, x, y):

for i in range(x):

if queue[i] == y or abs(x - i) == abs(queue[i] - y):

return True

return False

# 按列来摆放皇后

def put\_position(n, queue, col):

for i in range(n):

if not is\_attack(queue, col, i):

queue[col] = i

if col == n - 1: # 此时最后一个皇后摆放好了，打印结果。

print(queue)

else:

put\_position(n, queue, col + 1)

n = 4 # 这里是n 就是n皇后

queue = [None for i in range(n)] # 存储皇后位置的一维数组，数组下标表示皇后所在的列，下标对应的值为皇后所在的行。

put\_position(n, queue, 0)

Completeness：Yes

Optimality：Yes

Time complexly:最坏情况下o（n的n+1次幂）

Space complexly:o(n)