

## N-Queen problem

### Algorithm

Input:  $n \times n$  board and  $n$  queens

Output: Safe sequence of all queens on the board

### Algorithm:

```
if is_safe(matrix,0):
    #printing safe queen matrix
    print('after arranging',n,'queens board is')
    for i in range(len(matrix)):
        for j in range(len(matrix[1])):
            print(matrix[i][j],end=" ")
        print()
else:
    print('No possible safe sequence :( ')
```

### Recursive function for safe sequence:

```
def is_safe(matrix,n,posdiag,negdiag,column):
    if n>=len(matrix):
        return True
    for i in range(len(matrix)):
        if(posdiag[i+n]!=1 and negdiag[n-i]!=1 and column[i]!=1):
            matrix[i][n]=1
            posdiag[i+n]=negdiag[n-i]=column[i]=1
            if(is_safe(matrix,n+1)):
                return True

            matrix[i][n]=0
            posdiag[i+n]=negdiag[n-i]=column[i]=0
    return False
```

Code:

```
def is_safe(matrix,n):
    if n>=len(matrix):
        return True
    for i in range(len(matrix)):
        if(posdiag[i+n]!=1 and negdiag[n-i]!=1 and column[i]!=1):
            matrix[i][n]=1
            posdiag[i+n]=negdiag[n-i]=column[i]=1
            if(is_safe(matrix,n+1)):
                return True

            matrix[i][n]=0
            posdiag[i+n]=negdiag[n-i]=column[i]=0
    return False
```

```
n=int(input('n*n board : '))
matrix=[[0]*n for i in range(n)]
# matrix=[list(map(int,input().split())) for i in range(4) ]
(other way to input board)
column=[0]*len(matrix)
rpre=0
rcur=0
posdiag=[0]*(len(matrix)*len(matrix))

negdiag=[0]*(len(matrix)*len(matrix))

if is_safe(matrix,0):
    #printing safe queen matrix
    print('after arranging',n,'queens board is')
    for i in range(len(matrix)):
        for j in range(len(matrix[1])):
            print(matrix[i][j],end=" ")
        print()
else:
    print('No possible safe sequence :( ')
```

**Input and Output:**

```
PS E:\books and pdfs\sem4 pdfs\DAA lab> python .\n_queen.py
n*n board : 4
after arranging 4 queens board is
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0
PS E:\books and pdfs\sem4 pdfs\DAA lab> python .\n_queen.py
n*n board : 8
after arranging 8 queens board is
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0
PS E:\books and pdfs\sem4 pdfs\DAA lab> 
```