EFFICENT N QUEEN PROBLEM

NAME: J.L.PUGAZH MUKILAN

REG NO: 22BCE9292 SLOT :L7+L8

1)solve the n queen problem with least number of moves

```
import numpy as np

N = 4
valid_inputs = [1, 2, 3, 4]
matrix = np.zeros((N, N))
finalvalue = None
finalarray = None
firstime = True
efficency = 0
count = 1

def isSafe(matrix, row, col, num):
    # Check row
    for i in range(N):
        if matrix[row][i] == num:
            return False

    # Check column
    for i in range(N):
        if matrix[i][col] == num:
            return False

    return True
```

```python
def solvematrix(matrix, row, col, solutions):
    global efficency
    global count
    global finalvalue
    global firstime
    global finalarray
    efficency += 1
    if row == N - 1 and col == N:
        if firstime:
            finalvalue = efficency
            finalarray = np.copy(matrix)
            firstime = False
        else:
            if efficency < finalvalue:
                finalvalue = efficency
                finalarray = np.copy(matrix)

        efficency = 0
        return True

    # Move to the next row if column exceeds
    if col == N:
        row += 1
        col = 0

    # Skip filled cells
    if matrix[row][col] != 0:
        return solvematrix(matrix, row, col + 1, solutions)

    # Try filling the cell with each valid input
    for num in valid_inputs:
        if isSafe(matrix, row, col, num):
            matrix[row][col] = num
            solvematrix(matrix, row, col + 1, solutions)
            matrix[row][col] = 0  # Backtrack

    return False

solutions = {}
solvematrix(matrix, 0, 0, solutions)

if finalarray is not None:
    print("Final Matrix:")
    print(finalarray)
    print("Final Efficiency:", finalvalue)
else:
    print("No solution exists.")
```

OUTPUT:

```
C:\Users\Pugazh Mukilan\Desktop\SEM - 4\Artifical Intellgence - F2\LAB>"C:/P
rs/Pugazh Mukilan/Desktop/SEM - 4/Artifical Intellgence - F2/LAB/Efficent_N_
Final Matrix:
[[1. 2. 3. 4.]
 [2. 1. 4. 3.]
 [3. 4. 2. 1.]
 [4. 3. 1. 2.]]
Final Efficiency: 6
```