114. N-Queens Proble

Code:

```
def solve_n_queens(n):
    def is_valid(board, row, col):

        for i in range(row):
            if board[i] == col:
                return False

        for i, j in zip(range(row-1, -1, -1), range(col-1, -1, -1)):
            if board[i] == j:
                return False
        for i, j in zip(range(row-1, -1, -1), range(col+1, n)):
            if board[i] == j:
                return False
        return True

    def solve(row, board):
        if row == n:
            result.append(board[:])
            return
        for col in range(n):
            if is_valid(board, row, col):
                board[row] = col
                solve(row + 1, board)
                board[row] = -1
```

```
    result = []
    board = [-1] * n
    solve(0, board)
    return result


solutions = solve_n_queens(4)
for solution in solutions:
    for row in solution:
        print(' '.join('Q' if i == row else '.' for i in range(4)))
    print()
```

output:

Time complexity:f(n)=o(n!)