

EX.NO:1

DATE:11/9/2024

Reg.no:220701014

8- QUEENS PROBLEM

AIM : To implement an 8-Queens problem using Python.

You are given an 8x8 board; find a way to place 8 queens such that no queen can attack any other

queen on the chessboard. A queen can only be attacked if it lies on the same row, same column,

or the same diagonal as any other queen. Print all the possible configurations.

To solve this problem, we will make use of the Backtracking algorithm. The backtracking algorithm, in general checks all possible configurations and test whether the required result is

obtained or not. For the given problem, we will explore all possible positions the queens can be

relatively placed at. The solution will be correct when the number of placed queens = 8.



CODE:

```
def print_board(board):
    for row in board:
        print(' '.join('Q' if cell else '.' for cell in row))
    print()

def is_safe(board, row, col, N):
    # Check this column
    for i in range(row):
        if board[i][col]:
            return False

    # Check upper left diagonal
    for i, j in zip(range(row, -1, -1), range(col, -1, -1)):
        if board[i][j]:
            return False

    # Check upper right diagonal
    for i, j in zip(range(row, -1, -1), range(col, N)):
        if board[i][j]:
            return False

    return True

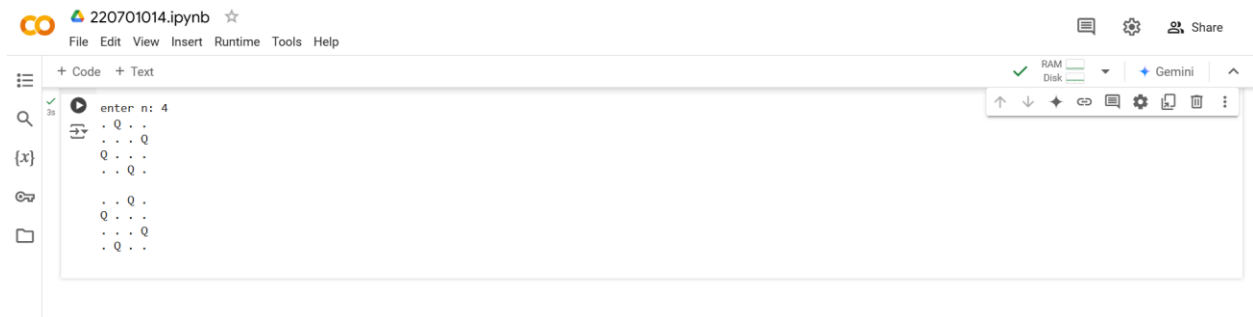
def solve_nqueens_util(board, row, N):
    if row >= N:
        print_board(board)
        return True

    res = False
    for col in range(N):
        if is_safe(board, row, col, N):
            board[row][col] = True
            res = solve_nqueens_util(board, row + 1, N) or res
            board[row][col] = False

    return res

def solve_nqueens(N):
    board = [[False] * N for _ in range(N)]
    if not solve_nqueens_util(board, 0, N):
        print("Solution does not exist")
N=int(input("enter n: "))
solve_nqueens(N)
```

OUTPUT:



The screenshot shows a Jupyter Notebook interface. At the top, there's a header with the Colab logo, the filename '220701014.ipynb', and a star icon. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right side of the header, there are icons for chat, settings, and a 'Share' button. The main area of the notebook is divided into two sections: a left sidebar with icons for file explorer, search, and other functions, and a central code editor. The code editor contains a single code cell with the following text: 'enter n: 4' followed by a loop that prints a pattern of dots. The output of the code cell is displayed below the code, showing the same pattern of dots. The pattern consists of 4 rows of 4 dots each, with the first row having a leading space. The output is formatted as a preformatted text block.

```
enter n: 4
. Q . .
. . . Q
Q . . .
. . Q .

. . Q .
Q . . .
. . . Q
. Q . .
```