```python
def n_queens(n):
    col = set()
    posDiag = set()  # (r+c)
    negDiag = set()  # (r-c)
    res = []
    board = [["0"] * n for i in range(n)]

    def backtrack(r):
        if r == n:
            copy = [" ".join(row) for row in board]
            res.append(copy)
            return

        for c in range(n):
            if c in col or (r + c) in posDiag or (r - c) in negDiag:
                continue
            col.add(c)
            posDiag.add(r + c)
            negDiag.add(r - c)
            board[r][c] = "1"
            backtrack(r + 1)
            col.remove(c)
            posDiag.remove(r + c)
            negDiag.remove(r - c)
            board[r][c] = "0"

    backtrack(0)

    for sol in res:
        for row in sol:
            print(row)
        print()

if __name__ == "__main__":
    n_queens(4)
```

```
0 1 0 0
0 0 0 1
1 0 0 0
0 0 1 0

0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0
```