

TahaAbbasALi_P200119_S23_AILab_09

April 12, 2023

```
[1]: import random

# chess board for 8 queens
def random_board(n):
    return [random.randint(0, n-1) for _ in range(n)]

def check_attacks(board):
    # the number of pairs of queens that can attack each other
    n = len(board)
    attacks = 0
    for i in range(n):
        for j in range(i+1, n):
            if board[i] == board[j] or abs(board[i]-board[j]) == j-i:
                attacks += 1
    return attacks

def hill_climbing(n):
    board = [random.randint(0, n-1) for _ in range(n)]
    # check_attacks of the initial configuration
    f = check_attacks(board)
    while True:
        best_board = None
        best_f = f
        for i in range(n):
            for j in range(n):
                if board[i] != j:
                    # small modification to the current configuration
                    new_board = list(board)
                    new_board[i] = j
                    new_f = check_attacks(new_board) # to evaluate the
    ↪ check_attacks of the modified configuration
                    if new_f < best_f:
                        best_board = new_board
                        best_f = new_f
        if best_f >= f:
            break
```

```

        else:
            board = best_board
            f = best_f
    return board, f

if __name__ == "__main__":
    n = 8
    attempts = 100
    successes = 0

    for _ in range(attempts):
        solution, h = hill_climbing(n)
        if h == 0:
            successes += 1

    success_rate = (successes / attempts) * 100
    print(f"Success rate for N = {8}: {success_rate}%")

    solution = hill_climbing(n)[0]
    print(solution, "\n")

arr = [[0 for j in range(8)] for i in range(8)]
# Place the list elements on the rows
for i, val in enumerate(solution):
    arr[val][i] = 1
for row in arr:
    print(row)

```

Success rate for N = 8: 23.0%

[3, 2, 7, 1, 4, 1, 5, 0]

[0, 0, 0, 0, 0, 0, 0, 1]

[0, 0, 0, 1, 0, 1, 0, 0]

[0, 1, 0, 0, 0, 0, 0, 0]

[1, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 1, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 1, 0]

[0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 1, 0, 0, 0, 0, 0]

[]: