Date / /
Page

LAB - 6

a) Implement A* search algorithm

b) Implement Hill climbing Algorithm for 8 Queens

-> a) A* (shortest path from initial point to the goal point)

- Firstly initialize the starting point as '1' and Assume goal state or end state to 6
- Visited and Unvisited nodes to be known.
- Then pick the nodes as priority queue and delete the nodes which is closer to the end node.

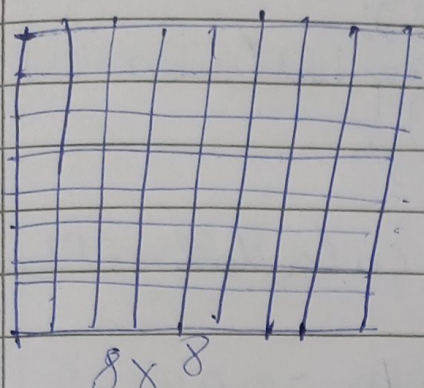
assume f(score) of the node
f(score) = g(score) + h(score)

g(score) -> actual distance travelled so far
h(score) -> Estimated distance left to reach the goal Node or End Node

- Then check the neighbour node and update the path
- Add neighbour state which doesn't get visited with a lower score before.
- Track the nodes to obtain the path which has Travelled.

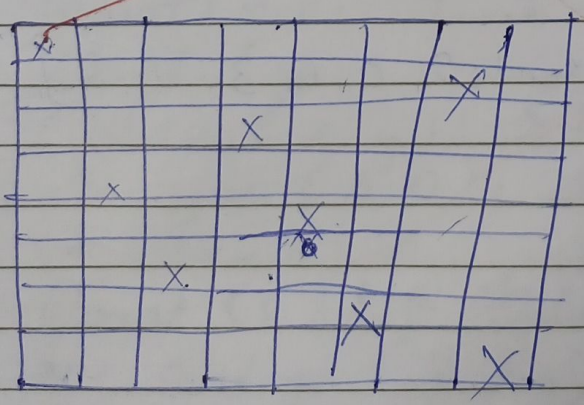
Date __/__/__
Page _____

Goal state: placing all queen on the board such that they cannot attack each other




```
def calculate_attacking_pair
attack = 0
n = len(board)
for i in range(n):
for j in range(i+1, n):
if board[i] == board[j] or
    abs(board[i] - board(j)) == abs(i-j):
attack += 1
return attacks
def a_star_8_queens(n=8):
open_set = priority Queue()
open_set_put([[0,1]])
for col in range(n):
    new_board = board + [col]
    g_score = len(new_board)
    h_score = calculate_attacking
pair(new_board)
f_score = g_score + h_score
```

The output will be



Date / /
Page

Lab 6

b) Implementing Hill climbing Algorithm & Queens

```
def hill_climbing(n=8):
    board = [random.randint(0, n-1)
             for i in range(n)]
    while True:
        current_attacks = calculate_attacking_
pairs(board)
        if current_attacks == 0:
            return board;
        for row in range(n):
            for col in range(n):
                if col != board[row]
```

- pair of queen attacking each other if attacks then return to the original state. if no check the position to for
- There should be no attacks for the node

Proceed

Output for A* algo:
Solution found: [0, 4, 7, 5, 2, 6, 1, 3]

Output for Hill climbing
Solution found:
[4, 1, 5, 0, 6, 3, 7, 2]

24/10/24