

Pooja Shrivastava
18M18CS069
AI Lab-1
13-11-2020
Pooja Shrivastava

a) Implemented the 8-puzzle problem using A* algorithm using heuristic function as Manhattan distance with depth not more than 3. If goal state is not reached with this limit, agent must report "NO SOLUTION"

8	2	3
	4	6
7	5	1

1	2	3
4	5	6
7	8	

1) Create the function that finds us the total estimated cost through node $n - f(n)$

```
// def print_grid(sec):
```

```
def h(state, target):
```

```
    // Manhattan distance
```

```
    dist = 0
```

```
    for i in state:
```

```
        d1, d2 = state.index(i), target.index(i)
```

```
        x1, y1 = d1 // 3, d1 % 3
```

```
        x2, y2 = d2 // 3, d2 % 3
```

```
        dist += abs(x1 - x2) + abs(y1 - y2)
```

```
    return dist
```

2) Create a search function

To traverse across the tree using $F(n)$ to select the next node

a) Make sure it to discard visited states

b) Create possible moves function

c) Create move generator function


```

def astar (sec, target):
    states = [sec]
    g = 0
    visited_states = set()
    while len(states):
        print (f"level: {g}")
        moves = []
        for state in states:
            visited_states.add (tuple (state))
            print_gid (state)
            if state == target:
                print ("Success")
                return
            moves += [move for move in possible_moves
                      (states, visited_states) if move
                      not in moves]
        costs = [g+h(move, target) for move in moves]
        states = [moves[i] for i in range (len(moves)) if
                  costs[i] == min(costs)]
        g += 1
    print ("NO SOLUTION")

```

```

// def possible_moves (state, visited_states):

```

```

// def gen (state, direction, b):

```