

Sakshi Srivastava

USN: 1BM18CS010

Batch - B3

12th Nov, 2020.

Sakshi

function :-

```
def astar(src, target):
```

```
    states = [src]
```

```
    g = 0
```

```
    visited_states = set()
```

```
    while len(states):
```

```
        print(f"Level: {g}")
```

```
        moves = []
```

```
        for state in states:
```

```
            visited_states.add(tuple(state))
```

```
            print_gnd(state)
```

```
            if state == target:
```

```
                print("Success")
```

```
                return
```

```
            moves += [move for move in possible_moves  
                      (state, 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):

 b = state.index(-1)

 d = []

 if q > b - 3 >= 0:

 d += 'u'

 if q > b + 3 >= 0:

 d += 'd'

 if b not in [2, 5, 8]:

 d += 'x'

 if b not in [0, 3, 6]:

 d += 'l'

 pos_moves = []

 for move in d:

 pos_moves.append(gen(state, move, b))

 return [move for move in pos_moves if tuple
 (move) not in visited_states]

def gen(state, direction, b):

 temp = state.copy()

if (direction == 'u'):

temp[b-3], temp[b] = temp[b], temp[b-3]

if (direction == 'd'):

temp[b+3], temp[b] = temp[b], temp[b+3]

if (direction == 'r'):

temp[b+1], temp[b] = temp[b], temp[b+1]

if direction == 'l':

temp[b-1], temp[b] = temp[b], temp[b-1]
return temp.

Manhattan distance.

def h(state, target):

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.

(3)