

13-Nov-2020

AI-LAB-1

Nitish.N. Banakar
18M18CS065

8 puzzle using A* search:-

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

```
    states = [src]
```

```
    g = 0
```

```
    visited_states = set()
```

```
    while len(states) and g <= depth:
```

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

```
        moves = []
```

```
        for state in states:
```

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

```
            print_grid(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]
```

```
            state = moves[i]
```

```
            for i in range(len(moves))
```

```
                if costs[i] == min(costs)]
```

```
            g += 1
```

```
    print("NO SOLUTION")
```

Nitish

```
def possible_moves(state, visited_states):
```

```
    b = state.index(-1)
```

```
    d = []
```

```
    if b >= 3:
```

```
        d += 'u'
```

```
    if b < 6:
```

```
        d += 'd'
```

```
    if (b+1) % 3 != 0
```

```
        d += 'r'
```

```
    if b % 3 != 0
```

```
        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, dir, b):
```

```
    temp = state.copy()
```

```
    if dir == 'u':
```

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

```
    elif dir == 'd':
```

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

```
    elif dir == 'r':
```

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

```
    else:
```

```
        temp[b-1], temp[b] = temp[b], temp[b-1]
```

```
    return temp
```

Handwritten signature

```

def print_grid (src):
    state = src.copy ()
    state[state.index(-1)] = ' '
    print( f"    {state[0]} {state[1]} {state[2]}
           {state[3]} {state[4]} {state[5]}
           {state[6]} {state[7]} {state[8]}
           """)

```

```

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

```

```

src = [ 8, 2, 3,
        -1, 4, 6,
        7, 5, 1 ]

```

```

target = [ 1, 2, 3,
           4, 5, 6,
           7, 8, -1 ]

```

```

depth = 3

```

```

astar (src, depth target, depth)

```

Nitin