07-Oct-2020

LAB-2

2. 8-puzzle problem pseudocode :-

```
def dfs (src, target, limit, visited_states):
    if src == target
        return true.

    if limit <= 0
        return false.
    visited_states.append (src)
    adj = possible_moves (src, visited_states)

    for move in adj:
        if dfs (move, target, limit-1, visited_states):
            return true.
    return False

def possible_moves (state, visited_states):

    ind = state.index (-1)
    d = []
    if ind +3 in range (9):
        d.append ('d')
    if ind-3 in range (9):
        d.append ('u')
    if ind not in [0,3,6]:
        d.append ('l')
    if ind not in [2,5,8]:
        d.append ('r')

    pos-moves = []
```

```
    for move in d:
        pos_moves.append(gen(state, move, ind))

    return [move for move in pos_moves if mow not
            in visited_states]


def gen(state, m, b):

    temp = state.copy()

    if m == 'd':
        a = temp[b+3]
        temp[b+3] = temp[b]
        temp[b] = a
    elif m == 'u'
        a = temp[b-3]
        temp[b-3] = temp[b]
        temp[b] = a
    elif m == 'l'
        a = temp[b-1]
        temp[b-1] = temp[b]
        temp[b] = a
    elif m == 'r'
        a = temp[b+1]
        temp[b+1] = temp[b]
        temp[b] = a
    return temp


def iddfs(src, target, depth):
    visited_states = []
```

```
for i in range(1, depth+1):
    if (dfs(src, target, i, visited_states):
        return True.
return False.


# Test 1
src = [1,2,3,-1,4,5,6,7,8]
target = [1,2,3,4,5,6,-1,7,8]
depth = 4

iddfs(src, target, depth)  # minimum depth should be 2.
```