8 Piece Puzzle Problem

```
def dfs(cur, target, limit, visited_states):
    if cur == target:
        return True
    if limit < 0:
        return False:
    visited_states.append(cur)
    moves = possible_moves(cur, visited_states)

    for move in moves:
        if dfs(move, target, limit-1, visited_states
            return True
    return False

def possible_moves(cur, 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')
    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(cur, move, ind))
    return [move for move in pos_moves if
                move not in visited_states]

def gen(state, move, 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
else:
    a = temp[b+1]
    temp[b+1] = temp[b]
    temp[b] = a
return temp

def iddfs(src, target, depth):
    visited_stated = []
    for i in range(1, depth+1):
        if dfs(src, target, i, visited_states):
            return True
    return False
```