24/11/2023 LPA 24-11-23

② 8-puzzle problem using Breadth-first search

```
def bfs(src, target):
    queue = []
    queue.append(src)
    #to keep track of already visited combination
    exp = []
    while len(queue) > 0:
        source = queue.pop(0)
        exp.append(source) # insert the already visited combination
        print(source) #output screen
        if source == target:
            print("Success")
            return
        #calculate the possible moves in the puzzle
        poss_moves_to_do = []
        poss_moves_to_do = possible_moves(source, exp)

        for moves in poss_moves_to_do:
            if move not in exp and move not in queue:
                queue.append(move)
```

index

0	1	2
3	4	5
6	7	8

2 3 4

```
def possible_moves(state, visited_states):
    b = state.index(0) // 6

    #direction array consists the moves that can be made
    d = []
    if b not in [0,1,2]:
        d.append('u')
    if b not in [6,7,8]:
        d.append('d')
    if b not in [0,3,6]:
        d.append('l')
    if b not in [2,5,8]:
        d.append('r')
    poss_moves_it_can = []
```

```
for i in d:        d['u','r']
    poss_moves_it_can.append(gen(state,i,b))
return [move_it_can for move_it_can in poss_move_it_can
        if move_it_can not in visited_states]

def gen(state, m, b):
    temp = state.copy()
    if m == 'd':
        temp[b+3], temp[b] = temp[b], temp[b+3]
    if m == 'u':
        temp[b-3], temp[b] = temp[b], temp[b-3]
    if m == 'l':
        temp['x'];
        temp[b-1], temp[b] = temp[b], temp[b-1]
    if m == 'r':
        temp[b+1], temp[b] = temp[b], temp[b+1]
    return temp
source = [1,2,3,4,5,6,0,7,8]
target = [1,2,3,4,5,6,7,8,0]
bfs(source, target)  # call this to print the sequence of moves
```

~~Algorithm~~:

Output:

```
1 | 2 | 3
4 | 5 | 6
0 | 7 | 8
-------

1 | 2 | 3
0 | 5 | 6
4 | 7 | 8
-------

1 | 2 | 3
4 | 5 | 6
7 | 0 | 8
-------

0 | 2 | 3
1 | 5 | 6
4 | 7 | 8
-------

1 | 2 | 3
5 | 0 | 6
4 | 7 | 8
-------

1 | 2 | 3
4 | 0 | 6
7 | 5 | 8
-------

1 | 2 | 3
4 | 5 | 6
7 | 8 | 0
-------
Success
```

```
1 | 2 | 3
4 | 5 | 6
0 | 7 | 8
-----------
1 | 2 | 3
0 | 5 | 6
4 | 7 | 8
-----------
1 | 2 | 3
4 | 5 | 6
7 | 0 | 8
-----------
0 | 2 | 3
1 | 5 | 6
4 | 7 | 8
-----------
1 | 2 | 3
5 | 0 | 6
4 | 7 | 8
-----------
1 | 2 | 3
4 | 0 | 6
7 | 5 | 8
-----------
1 | 2 | 3
4 | 5 | 6
7 | 8 | 0
-----------
Success
```