WEEK 3

8 PUZZLE PROBLEM

def bfs(src,target):

    queue = []

    queue.append(src)

    exp = []

    while len(queue) > 0:

        source = queue.pop(0)

        exp.append(source)

        print(source)

        if source==target:

            print("success")

            return

        poss\_moves\_to\_do = []

        poss\_moves\_to\_do = possible\_moves(source,exp)

        for move in poss\_moves\_to\_do:

            if move not in exp and move not in queue:

                queue.append(move)

def possible\_moves(state,visited\_states):

    #index of empty spot

    b = state.index(-1)

    #directions array

    d = []

    #Add all the possible directions

    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')

    # If direction is possible then add state to move

    pos\_moves\_it\_can = []

    # for all possible directions find the state if that move is played

    ### Jump to gen function to generate all possible moves in the given directions

    for i in d:

        pos\_moves\_it\_can.append(gen(state,i,b))

    return [move\_it\_can for move\_it\_can in pos\_moves\_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[b-1],temp[b] = temp[b],temp[b-1]

    if m=='r':

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

    # return new state with tested move to later check if "src == target"

    return temp

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

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

bfs(src, target)

OUTPUT:

