def dfs(src,target,limit,visited\_states):

if src == target:

return True

if limit <= 0:

return False

visited\_states.append(src)

moves = possible\_moves(src,visited\_states)

for move in moves:

if dfs(move, target, limit-1, visited\_states):

return True

return False

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

b = state.index(-1)

d = []

if b not in [0,1,2]:

d += 'u'

if b not in [6,7,8]:

d += 'd'

if b not in [2,5,8]:

d += 'r'

if b not in [0,3,6]:

d += 'l'

pos\_moves = []

for move in d:

pos\_moves.append(gen(state,move,b))

return [move for move in pos\_moves if move not in visited\_states]

def gen(state, move, blank):

temp = state.copy()

if move == 'u':

temp[blank-3], temp[blank] = temp[blank], temp[blank-3]

if move == 'd':

temp[blank+3], temp[blank] = temp[blank], temp[blank+3]

if move == 'r':

temp[blank+1], temp[blank] = temp[blank], temp[blank+1]

if move == 'l':

temp[blank-1], temp[blank] = temp[blank], temp[blank-1]

return temp

def iddfs(src,target,depth):

for i in range(depth):

visited\_states = []

if dfs(src,target,i+1,visited\_states):

return True

return False

#Test 1

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

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

depth = 1

iddfs(src, target, depth)

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

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

for i in range(1, 100):

val = iddfs(src,target,i)

print(i, val)

if val == True:

break

OUTPUT:

1 False

2 False

3 False

4 False

5 False

6 False

7 False

8 False

9 False

10 False

11 False

12 False

13 False

14 False

15 False

16 False

17 False

18 False

19 False

20 False

21 False

22 False

23 False

24 False

25 True