WEEK-3

8 -PUZZLE PROBLEM USING BFS

import numpy as np

import pandas as pd

import os

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):

b = state.index(0)

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')

pos\_moves\_it\_can = []

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 temp

SAMPLE OUTPUT:





