1. write the python to solve 8 puzzle problem

import heapq

goal = [[1,2,3],[4,5,6],[7,8,0]]

moves = [(-1,0),(1,0),(0,-1),(0,1)]

def dist(b): return sum(abs(i-(v-1)//3)+abs(j-(v-1)%3) for i in range(3) for j in range(3) if (v:=b[i][j]))

def find\_zero(b): return next((i,j) for i in range(3) for j in range(3) if b[i][j]==0)

def next\_b(b):

x,y=find\_zero(b);res=[]

for dx,dy in moves:

nx,ny=x+dx,y+dy

if 0<=nx<3 and 0<=ny<3:

nb=[r[:] for r in b]

nb[x][y],nb[nx][ny]=nb[nx][ny],nb[x][y]

res.append(nb)

return res

def solve(start):

heap=[(dist(start),0,start,[])]; seen=set()

while heap:

\_,s,b,path=heapq.heappop(heap)

if b==goal: return path+[b]

k=str(b)

if k in seen: continue

seen.add(k)

for nb in next\_b(b): heapq.heappush(heap,(s+1+dist(nb),s+1,nb,path+[b]))

start = [[1,2,3],[4,0,6],[7,5,8]]

for i,step in enumerate(solve(start)):

print(f"Step {i}:"); [print(r) for r in step]; print()

OUTPUT:

