Python program for Greedy Best First Search

import heapq

goal = '123456780'

moves = {0:[1,3],1:[0,2,4],2:[1,5],3:[0,4,6],4:[1,3,5,7],5:[2,4,8],6:[3,7],7:[4,6,8],8:[5,7]}

def h(s): return sum(a!=b and a!='0' for a,b in zip(s,goal))

def greedy(start):

pq = [(h(start), start, "")]

visited = {start}

while pq:

\_, s, path = heapq.heappop(pq)

if s == goal: return path

i = s.index('0')

for m in moves[i]:

t = list(s)

t[i], t[m] = t[m], t[i]

ns = ''.join(t)

if ns not in visited:

visited.add(ns)

heapq.heappush(pq, (h(ns), ns, path + ns + " "))

return "No solution"

start = '103425786'

print("Steps:\n", greedy(start))

Output

Steps:

123405786 123450786 123456780