Week 4

Implement 8 puzzle using A\* algorithm

def print\_in\_format(matrix):

for i in range(9):

if i%3==0 and i>0:

print("")

print(str(matrix[i])+" ", end = "")

def convert(s):

mat = []

a = []

b = []

c = []

for i in range(9):

if i<3:

a.append(s[i])

if i>=3 and i<=5:

b.append(s[i])

if i>5:

c.append(s[i])

mat.append(a)

mat.append(b)

mat.append(c)

return mat

def ideal\_distFind(val):

x1 = 999

y1 = 999

ideal = [[1, 2, 3],

[4, 5, 6],

[7, 8, 0]]

for i in range(3):

for j in range(3):

if ideal[i][j]==val:

x1 = i

y1 = j

break

return x1, y1

def count(initial\_state):

inits = initial\_state.copy()

inicon = convert(inits)

x1 = y1 = x2 = y2 = 999

total\_h = 0;

for i in range(3):

for j in range(3):

x1, y1 = ideal\_distFind(inicon[i][j])

x2, y2 = i, j

total\_h += abs(x1-x2)+abs(y1-y2)

return total\_h

def move(ar, p, st):

rh = 9999

store\_st = st.copy()

for i in range(len(ar)):

dupl\_st = st.copy()

tmp = dupl\_st[p]

dupl\_st[p] = dupl\_st[arr[i]]

dupl\_st[arr[i]] = tmp

trh = count(dupl\_st)

if trh<rh:

rh = trh

store\_st = dupl\_st.copy()

#print(rh, store\_st)

return store\_st, rh

state = [0, 1, 3,

4, 2, 5,

7, 8, 6]

h = count(state)

Level = 1

print("\n------ Level "+str(Level)+" ------")

print\_in\_format(state)

print("\nHeuristic Value(Manhattan Distance) : "+str(h))

while h>0:

pos = int(state.index(0))

Level += 1

if pos==0:

arr = [1, 3]

state, h = move(arr, pos, state)

elif pos==1:

arr = [0, 2, 4]

state, h = move(arr, pos, state)

elif pos==2:

arr = [1, 5]

state, h = move(arr, pos, state)

elif pos==3:

arr = [0, 4, 6]

state, h = move(arr, pos, state)

elif pos==4:

arr = [1, 3, 5, 7]

state, h = move(arr, pos, state)

elif pos==5:

arr = [2, 4, 8]

state, h = move(arr, pos, state)

elif pos==6:

arr = [3, 7]

state, h = move(arr, pos, state)

elif pos==7:

arr = [4, 6, 8]

state, h = move(arr, pos, state)

elif pos==8:

arr = [5, 6]

state, h = move(arr, pos, state)

print("\n------ Level "+str(Level)+" ------")

print\_in\_format(state)

print("\nHeuristic Value(Manhattan Distance) : "+str(h))

Output:

