**Assignment No 3**

**Name : Aniket B. Suryawanshi**

**Roll No : B-46**

**Subject : AI**

**CS324.1: Formulate and solve sequence of actions for an agent as a search problem.**

**PROBLEM STATEMENT**

**Implement a suitable search strategy to solve an 8-puzzle problem.**

**INSTRUCTIONS**

**Implement any one type of uniformed search strategy. Give the reason for**

**using the strategy that you have chosen.**

**Code :**

import time

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

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

c = 0 #66518339

start = time.time()

def recurse(m,i,j,k):

if m == g:

print(m)

print("Got the state")

global c

print(c)

end = time.time()

print("Execution time is ",end-start)

exit(0)

if k>15:

return

print(m)

c+=1

if j+1<3:

a = [row[:] for row in m]

a[i][j],a[i][j+1] = a[i][j+1],a[i][j]

recurse(a,i,j+1,k+1)

if i+1<3:

a = [row[:] for row in m]

a[i][j], a[i + 1][j] = a[i + 1][j], a[i][j]

recurse(a, i + 1, j,k+1)

if i-1 >=0 :

a = [row[:] for row in m]

a[i][j], a[i - 1][j] = a[i - 1][j], a[i][j]

recurse(a, i - 1, j,k+1)

if j - 1>=0:

a = [row[:] for row in m]

a[i][j], a[i][j - 1] = a[i][j - 1], a[i][j]

recurse(a, i, j - 1,k+1)

recurse(m,1,0,1)

**Theory :**

We used depth limit search because it is easy to implement and

we can limit the depth by k value and

it will take almost same time as BFS if we

decide the k value correctly.

**Output :**

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

Got the state

77

Execution time is 0.0010020732879638672