AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Department of Computer Science and Engineering

Program: Bachelor of Science in Computer Science and Engineering
Spring 2019

Course No: CSE 4108

Course Title: Artificial Intelligence Lab

Term Project No: 02

Topic: First Choice Hill Climbing

Date of Submission: 6 October, 2019

Submitted To: Dr. S.M. Abdullah Al-Mamun
Professor, Department of CSE, AUST.

Md. Siam Ansary
Part time Faculty, Department of CSE, AUST.

Submitted By,

Name: Iffatun Nessa
Student ID: 160104031
Group: A2

First-choice hill climbing implements hill climbing by generating successors, and choose one that is better than the current state. This is a good strategy when a state has many (e.g., thousands) of successors. So First-choice hill climbing is a special kind of stochastic hill climbing.

Input:

A number will be taken as a choice, whether the user wants to clear, execute the procedure, display, save the state or exit.

If the user chooses executing the procedure then s/he has to input the state and the threshold value.

Major steps:

1. find the heuristic value of a state and execute 28 – heuristic value as a fitness value.

2. generates 56 successors for the current state and find the fitness value for all successor.

3. if the first generated successor does not have better value than current state then discard it and continue the procedure for other successors.

4. else, take that state as current state and find the successor for that state.

5. repeat step 3 and 4 until the threshold value is not found.

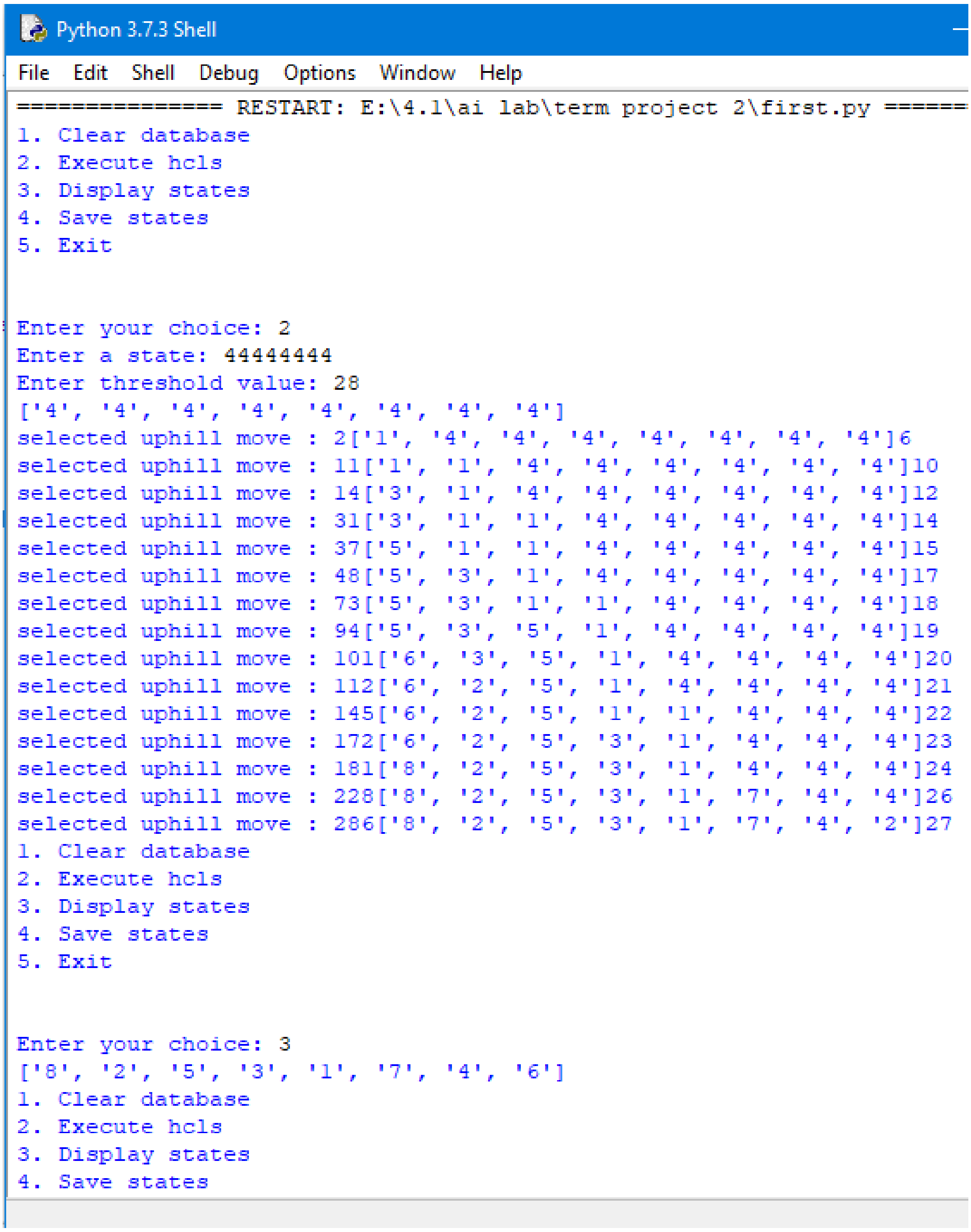
Implementation in python:

```
threshold = 0

goal=[]

#state=[4,4,4,4,4,4,4,4]

iteration=0

def clear_database():

   global threshold

   global  state

   global iteration


   threshold = 0

   state = []

   iteration = 0


def execute_firstchoice_hc():

   global threshold
```

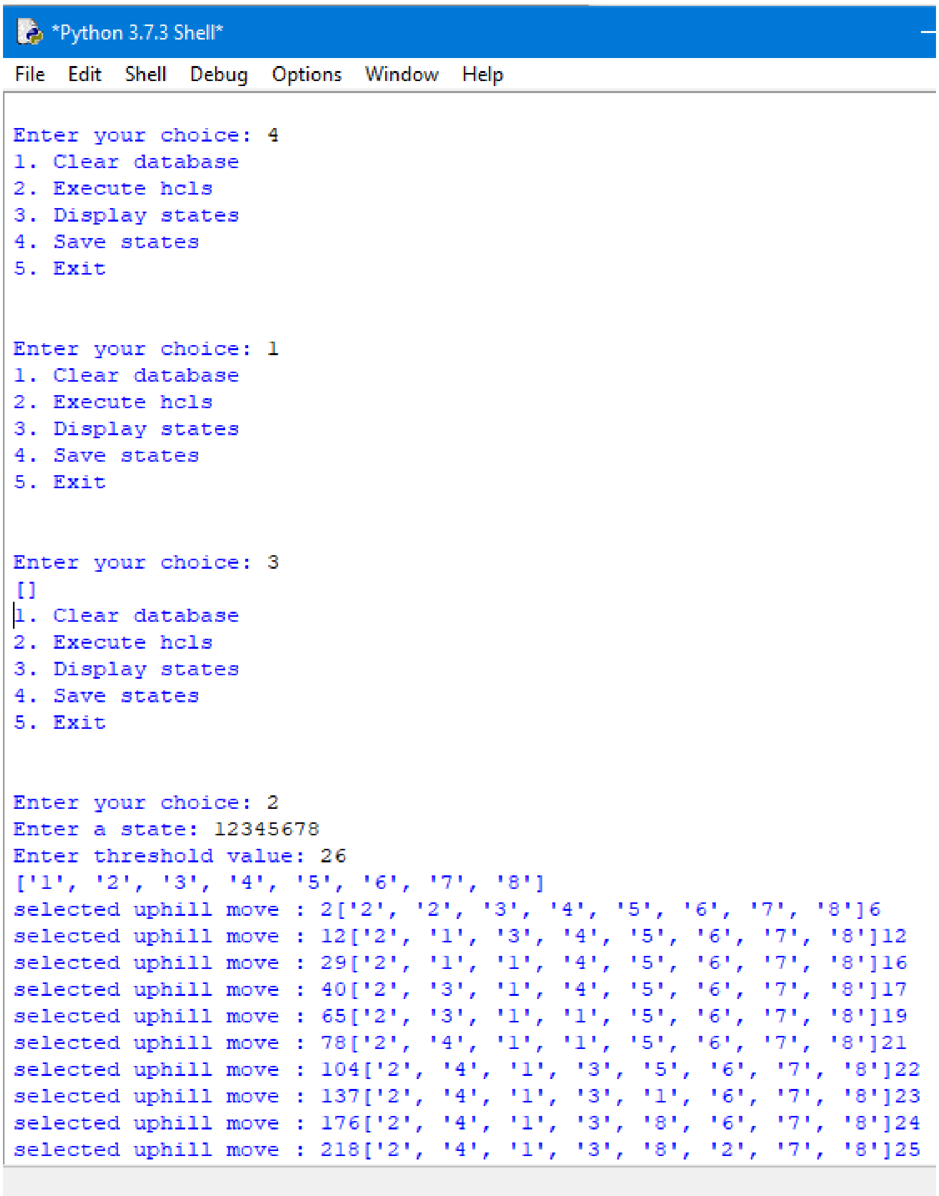
```
    global  state
    global iteration

    s = input('Enter a state: ')
    threshold = int(input('Enter threshold value: '))
    state=list(s)
    print(state)
    firstchoice(state)

def display_states():
    global state
    print(state)

def save_states():
    global state
    f1=open("std.py", "w")
    print(state,file = f1)
    f1.close

def evaluate_nonattacking(s):
    count = 0
    for i in range(0,len(s),1):
        for j in range(i+1,len(s),1):
            if (s[i] == s[j]):
                count = count + 1

    for i in range(0,len(s),1):
        x = int(s[i])
        for j in range(i+1,len(s),1):


            x = x + 1
            if (x == int(s[j])):
                count = count + 1

    for i in range(0,len(s),1):
        x = int(s[i])
        for j in range(i+1,len(s),1):
            x = x - 1
            if (x == int(s[j])):
                count = count + 1

    #print(count)
    return 28 - count

def firstchoice(state):
    global iteration
    i=0
    while i<8:
        temp = state[i]
        hval = 0
        current_hval = evaluate_nonattacking(state)
        better = False
        for j in range(1,9):
            if(state[i] == str(j)):
                continue
            state[i]=str(j)
            iteration+=1;
            hval=evaluate_nonattacking(state)
            if(hval == threshold):
                goal.append((iteration, state,hval))
                return
            if(hval>current_hval):
```

```
            iteration +=1
            i=0
            print('selected uphill move :',str(iteration)+str(state)+str(hval))
            better = True
            break
      if not better:
         state[i] = temp
         i+=1
   if goal:
      print("Goal:"+str(goal_state[0][0])+ 'state:'+str( goal[0][1])+ 'val:'+str(goal[0][2]))
      print()
   else: print("Stuck")

#Main
count = 1

while(count>=1 and count<5):
   print('1. Clear database')
   print('2. Execute hcls')
   print('3. Display states')
   print('4. Save states')
   print('5. Exit')
   cs = int(input('\n\nEnter your choice: '))
   if(cs == 1):
      clear_database()
   elif(cs == 2):
      execute_firstchoice_hc()
      #firstchoice(state)
   elif(cs == 3):
      display_states()
   elif(cs == 4):
      save_states()
   else:
      break
```

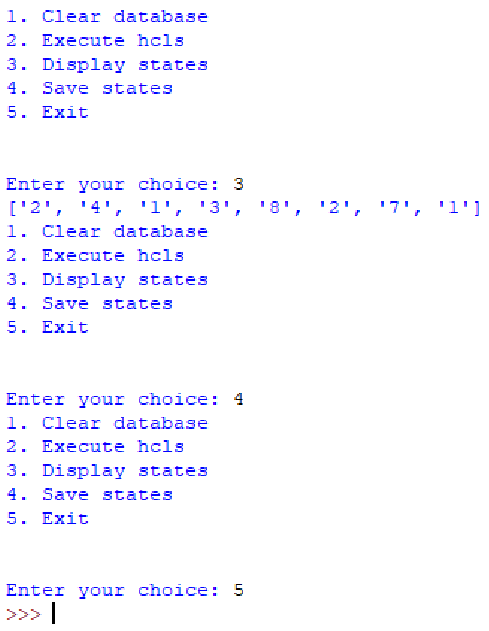
Output:



```
=============== RESTART: E:\4.1\ai lab\term project 2\first.py ======
1. Clear database
2. Execute hcls
3. Display states
4. Save states
5. Exit


Enter your choice: 2
Enter a state: 44444444
Enter threshold value: 28
['4', '4', '4', '4', '4', '4', '4', '4']
selected uphill move : 2['1', '4', '4', '4', '4', '4', '4', '4']6
selected uphill move : 11['1', '1', '4', '4', '4', '4', '4', '4']10
selected uphill move : 14['3', '1', '4', '4', '4', '4', '4', '4']12
selected uphill move : 31['3', '1', '1', '4', '4', '4', '4', '4']14
selected uphill move : 37['5', '1', '1', '4', '4', '4', '4', '4']15
selected uphill move : 48['5', '3', '1', '4', '4', '4', '4', '4']17
selected uphill move : 73['5', '3', '1', '1', '4', '4', '4', '4']18
selected uphill move : 94['5', '3', '5', '1', '4', '4', '4', '4']19
selected uphill move : 101['6', '3', '5', '1', '4', '4', '4', '4']20
selected uphill move : 112['6', '2', '5', '1', '4', '4', '4', '4']21
selected uphill move : 145['6', '2', '5', '1', '1', '4', '4', '4']22
selected uphill move : 172['6', '2', '5', '3', '1', '4', '4', '4']23
selected uphill move : 181['8', '2', '5', '3', '1', '4', '4', '4']24
selected uphill move : 228['8', '2', '5', '3', '1', '7', '4', '4']26
selected uphill move : 286['8', '2', '5', '3', '1', '7', '4', '2']27
1. Clear database
2. Execute hcls
3. Display states
4. Save states
5. Exit


Enter your choice: 3
['8', '2', '5', '3', '1', '7', '4', '6']
1. Clear database
2. Execute hcls
3. Display states
4. Save states
```



```
Enter your choice: 4
1. Clear database
2. Execute hcls
3. Display states
4. Save states
5. Exit


Enter your choice: 1
1. Clear database
2. Execute hcls
3. Display states
4. Save states
5. Exit


Enter your choice: 3
[]
1. Clear database
2. Execute hcls
3. Display states
4. Save states
5. Exit


Enter your choice: 2
Enter a state: 12345678
Enter threshold value: 26
['1', '2', '3', '4', '5', '6', '7', '8']
selected uphill move : 2['2', '2', '3', '4', '5', '6', '7', '8']6
selected uphill move : 12['2', '1', '3', '4', '5', '6', '7', '8']12
selected uphill move : 29['2', '1', '1', '4', '5', '6', '7', '8']16
selected uphill move : 40['2', '3', '1', '4', '5', '6', '7', '8']17
selected uphill move : 65['2', '3', '1', '1', '5', '6', '7', '8']19
selected uphill move : 78['2', '4', '1', '1', '5', '6', '7', '8']21
selected uphill move : 104['2', '4', '1', '3', '5', '6', '7', '8']22
selected uphill move : 137['2', '4', '1', '3', '1', '6', '7', '8']23
selected uphill move : 176['2', '4', '1', '3', '8', '6', '7', '8']24
selected uphill move : 218['2', '4', '1', '3', '8', '2', '7', '8']25
```



```
1. Clear database
2. Execute hcls
3. Display states
4. Save states
5. Exit


Enter your choice: 3
['2', '4', '1', '3', '8', '2', '7', '1']
1. Clear database
2. Execute hcls
3. Display states
4. Save states
5. Exit


Enter your choice: 4
1. Clear database
2. Execute hcls
3. Display states
4. Save states
5. Exit


Enter your choice: 5
>>> 
```