CS210: ARTIFICIAL INTELLIGENCE LAB

LAB ASSIGNMENT 2: AI & Python

Submitted By:

Name: Akshat Sahu

Roll No: U22CS034

Branch: CSE

Semester: 4th Sem

Division : A

Submitted To: Dr. Chandra Prakash

Department of Computer Science and Engineering



SV NATIONAL INSTITUTE OF TECHNOLOGY SURAT

2024

1	
PAGE NO.	
TATE:	

	PAGE NO.
	PART = YL 8 + WOS
(4	
	AUTONOMOUSE ROBOT FOROM HUDNOCUILS INDICT * 1
11	Ceneous & Ceneous we the object in Hosof that
	an House and information around its
	RUNHAUNDING SO that next action can be taken based on that information
	based on that intermation promot
	MOMANDE SINGENETIC K
+	Centory that we use in and Hobotson
	HADDON, MOHOUS Y
•	Sonar Senserus => 9+ is used to devert objects
	and distance between them
	mobot and that object. Motor singles +
	HOTOM OUTURE Y
•	haven fencon => 94 is used measured distance
	between the object & and rubor.
	Puddlotic Achanan
•	Visitoal Sonson => 9t is ored to identify surroording V= 10000 biomon Environments of 99t2
	Y= HORDY BLONDY ENVIRONMENTED OF 95972 Y
1	Octobra de la Constante de Cons
2)	Actuators = 4 Actuators are the components that
	to the partour the action on the basic
a treets as	of the information provided by the gonears.
1	MOTON = 3+ is used ton the mechanical motion of
1)	the HODOF. THE MENTINGER MOHON of
	THE GOODE PROPERTY ASSESSMENT AS
2)	Brake = 4 It is used to control the motion.
	Si vs ogen in control in biomari
3	exercing to 9+ nelps in controlling the direction.
1	
4	Hydraulic Acwatone > 9t is used to deliver High
	MINON
	priximity 0 (2.

```
import sys
def is room clean(dirt status):
    return all(d == 0 for d in dirt_status)
def vacuum(room_number, dirt_status, algorithm):
   valid_actions = ['L', 'R', 'U', 'D', 'S', 'N']
   if algorithm == 'dfs':
        stack = [(room_number, [])]
   elif algorithm == 'bfs':
        queue = [(room_number, [])]
   else:
        print("Invalid algorithm specified.")
        return
   while stack or queue:
        if algorithm == 'dfs':
            current_room, actions = stack.pop()
        elif algorithm == 'bfs':
            current_room, actions = queue.pop(∅)
        if is room clean(dirt status):
            print_sequence(actions)
            return
        for action in valid actions:
            new_room, new_dirt_status = simulate_action(current_room,
dirt_status, action)
            if algorithm == 'dfs':
                stack.append((new_room, actions + [(current_room, action)]))
            elif algorithm == 'bfs':
                queue.append((new_room, actions + [(current room, action)]))
    print("Goal state not reached.")
def simulate_action(current_room, dirt_status, action):
    new_room = current_room
   new_dirt_status = dirt_status.copy()
   return new_room, new_dirt_status
```

```
def print_sequence(actions):
    for room, action in actions:
        print(f"{room},{action}")

def main():
    if len(sys.argv) != 3:
        print("Invalid number of arguments.")
        return

    input_file = sys.argv[1]
    output_file = sys.argv[2]

    with open(input_file, "r") as file:
        room_number = int(file.readline().strip())
        dirt_status = list(map(int, file.readline().strip().split(',')))
        algorithm = file.readline().strip()
        vacuum(room_number, dirt_status, algorithm)

if __name__ == "__main__":
    main()
```

3:

```
from collections import defaultdict
class Graph:
   def __init__(self):
       self.adjacency_list = defaultdict(list)
   def add_edge(self, u, v):
       self.adjacency_list[u].append(v)
       self.adjacency_list[v].append(u)
   def bfs(self, start, goal):
       visited = [False] * len(self.adjacency_list)
       queue = []
       path = []
       queue.append(start)
       visited[start] = True
       while queue:
           current = queue.pop(0)
           path.append(current)
```

```
if current == goal:
                return path
            for neighbor in self.adjacency_list[current]:
                if not visited[neighbor]:
                    queue.append(neighbor)
                    visited[neighbor] = True
   def dfs(self, current, goal, visited, path):
       path.append(current)
       visited[current] = True
       if current == goal:
            return path
       for neighbor in self.adjacency_list[current]:
            if not visited[neighbor]:
               return self.dfs(neighbor, goal, visited, path)
def main():
   g = Graph()
   g.add_edge(1, 2)
   g.add_edge(2, 3)
   g.add_edge(3, 4)
   g.add_edge(3, 5)
   g.add_edge(5, 4)
   start_state = int(input("Enter starting state (1-5): "))
   goal_state = int(input("Enter state (1-5): "))
   approach = input("Enter approach (BFS/DFS): ")
   if start_state < 1 or start_state > 5 or goal_state < 1 or goal_state > 5:
       print("Invalid input.")
   if approach.lower() == 'bfs':
       bfs_path = g.bfs(start_state, goal_state)
       if bfs_path:
            print("BFS Path:", "-".join(map(str, bfs_path)))
       else:
            print("Path not found.")
   elif approach.lower() == 'dfs':
       visited = [False] * len(g.adjacency_list)
       dfs_path = g.dfs(start_state, goal_state, visited, [])
```

PART B

1:

4.	
1000	PAGE NO.
	Cour.
	PONT B
The same	
1 *	Tonyli Hiwondon Robot 19009 DUMONOROL
(t which	Poxcieving dovices and Actuatory
4	michophone o 1x01 to 4 2x on the state of th
(POY)	michophone of two to the or productions
7	COMENA MOLLONIC SONOLON HOSPING SINGLONATION
- 	Ulthalonic songon
=	Touch Leneous 19 19 19 19 19 19 19 19 19 19 19 19 19
4	MOHON LOWOH
210010	0 - 10 910 07 bigo 21 48 4 24609102 HONOZ 4
2)	# Action sometable some
. 4	POUND MATOU
KEUNCK .	BHORE SO OH NORTH SHOW .
suday, 7	OND BHORE TO OHE MONDERSH
7	hydraotic Acharan.
SONAMORANOS	Historia of post of the Land of motion of
-	Lydraotic Achatan. Stepe to build a Momanoid Robot=4
LOH 91	Planning =44 040) lungtons (c) 445 primall
MOD Y	Making a note of functionality of Kobot
213210	Making a note of functionality of Robot Making a design based on the functionality you not
7	Maring a assign based on the functionality you never
TO MOHOM OF	COMPAN CONTRACT DOZIO ST TO PROTOM (I
2)	CONCLOSE ACCOMBLY 1000000 ON
0	Paragraphical haller of the V
NOHS	2) Ruape - 4 3t 12 10god - to controllange
1.2	Disposition in the state of the
4)	Browning - 4 St valor in controlle (8
18.11	Parant finally by the manager of city
MOIH 5	Power Lopping 21 +8 & Wordensh Silonahun (1)
6	Optimizing.
6)	obunia.

-	PAGE NO.
	ONE:
2)*	Taxtle Bot 3
	प्रचारकार १=
	Penciving devices wower shoult
	J. Hadenbring +
0	Cliff sensor
2) 3)	in extial mealiprement shows to
3)	Camera.
re NOM	tidanos Lensano 291) ogo 1909, vol proposo Alemonos 900 A
	URE OF the analyse though wotion.
1)	Hewator
2)	Dynamixel motors
9)	DAMENT PROTOGO
->	Steps to boild an Aptonomaus Robot.
1)	Planning
2)	Hardware Assembly
3)	Pawer Supply
4	Handware Assembly fawan Supply Solfall tunksbot 3 Parages Cersony Calibration
5)	Censory Calibration
6)	Mapping SULP
7	Laonen Mapping noons
8) \$)	Mapping Ellup Laonan Mapping nodes Map visualization Save map Optimization
	Suit Map
10)	Optim range.
*	general in mobile devices which of them could
-	be osed in Human Motion analysis.
	The state of the s
· -	Acceleromeren
-4	SUNOSCOPE
=4	Magnetometer
	J

	PAGE NO. DATE:
+	GPC E 409 STATES
	BHOSCIMITY & C
=	Light Remore 2901000 DVIVIDAG
7	Fingerpuint
=	MOZNOS (MI)
4	Camera medicinement
	HECCELEROMETER, GUROCCOPE, GRC and camera con
	Accelerometer, Gyrolcope, GRS and lamera earth
	Rejoot A
	2002/4
	LK9111
	MOTOM 19XIMINET 8
	- 19das to boild an Aprenomans Rabet
	100 100 Ot 24987 1
	Martinaga Assombly !
	Charle Manual
	Lopingua & tod Distate to the second of the
	NEW DWOLLD'S AUGUST STATES
	The state of the s
	Albor principly bronois
	MOREO TOUR SUPER CONTROL
	JOHN Start
bloop	Malatica
43.50	10 allaida de la
	The sound in monday monday and the sound of
	OF 0300 IN EDGE
, who were	
	* HOELPHOMEHEM
1	MOTORNOSTA