

MI Assignment - Week 2

Name : Adarsh Subhas Nayak

SRN : PES1UG20CS620

Roll No : 54

Date : 18-08-2022

Code :

```
PES1UG20CS620.py U x
C: > Users > Hp > Desktop > Machine Intelligence > PES1UG20CS620 > Lab > Week 2 > PES1UG20CS620.py > DFS_Traversal
2 You can create any other helper functions.
3 Do not modify the given functions
4 """
5 import queue
6 import copy
7
8 def A_star_Traversal(cost, heuristic, start_point, goals):
9     n = len(cost)
10    visited = [0 for i in range(n)]
11    fpq = queue.PriorityQueue()
12    fpq.put((heuristic[start_point], ([start_point], start_point, 0)))
13    while(fpq.qsize() != 0):
14        tec, node_tup = fpq.get()
15        A_star_path_till_node = node_tup[0]
16        node = node_tup[1]
17        node_cost = node_tup[2]
18        if visited[node] == 0:
19            visited[node] = 1
20            if node in goals:
21                return A_star_path_till_node
22            for neighbour_node in range(1, n):
23                if cost[node][neighbour_node] > 0 and visited[neighbour_node] == 0:
24                    tot_cost_node = node_cost + cost[node][neighbour_node]
25                    est_tot_cost = tot_cost_node + heuristic[neighbour_node]
26                    star_path_till_n = copy.deepcopy(A_star_path_till_node)
27                    star_path_till_n.append(neighbour_node)
28                    fpq.put((est_tot_cost, (star_path_till_n, neighbour_node, tot_cost_node)))
29    return list()
30
```

```
PES1UG20CS620.py u x
C: > Users > Hp > Desktop > Machine Intelligence > PES1UG20CS620 > Lab > Week 2 > PES1UG20CS620.py > ...

33
34 def DFS_Traversal(cost, start_point, goals):
35     """
36     Perform DFS Traversal and find the optimal path
37     cost: cost matrix (list of floats/int)
38     start_point: Starting node (int)
39     goals: Goal states (list of ints)
40     Returns:
41     path: path to goal state obtained from DFS(list of ints)
42     """
43     path = []
44     # TODO
45     front = []
46     n = len(cost)
47     front.append(start_point)
48     while len(front) != 0:
49         curr = front.pop()
50         path.append(curr)
51         if curr in goals:
52             return path
53         for i in range(n - 1, 0, -1):
54             if cost[curr][i] != -1 and cost[curr][i] != 0 and (i not in path):
55                 front.append(i)
56     return path
57
58
59
60
```

main* Run Testcases 0 0 0 Ln 6, Col 12 Spaces: 4 UTF-8 CRLF Python 3.10.1 64-bit Go Live

Output :

```
Select C:\Windows\System32\cmd.exe
C:\Users\Hp\Desktop\Machine Intelligence\PES1UG20CS620\Lab\Week 2>python SampleTest.py --SRN PES1UG20CS620
Test Case 1 for A* Traversal PASSED
Test Case 2 for DFS Traversal PASSED

C:\Users\Hp\Desktop\Machine Intelligence\PES1UG20CS620\Lab\Week 2>
```