MACHINE INTELLIGENCE LABORATORY

WEEK – 2

NAME : SANMAT SANJAYAKUMAR PAYAGOUDAR

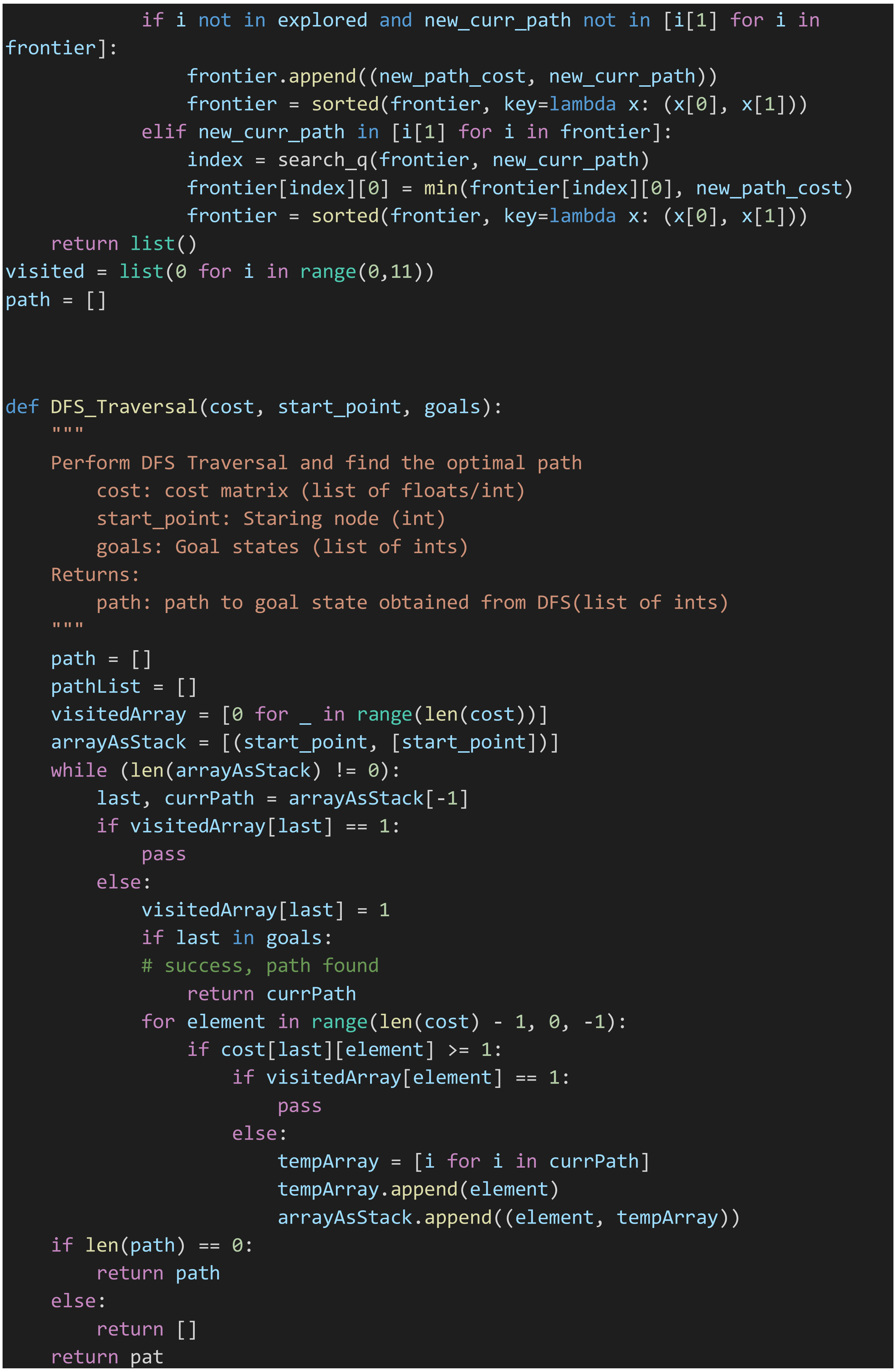
SRN : PES1UG20CS385

CODE :

PES1UG20CS385.py

```python
"""
You can create any other helper funtions.
Do not modify the given functions
"""


def A_star_Traversal(cost, heuristic, start_point, goals):
    """
    Perform A* Traversal and find the optimal path
    Args:
        cost: cost matrix (list of floats/int)
        heuristic: heuristics for A* (list of floats/int)
        start_point: Staring node (int)
        goals: Goal states (list of ints)
    Returns:
        path: path to goal state obtained from A*(list of ints)
    """
    path = []
    explored = []
    path = [start_point]
    frontier = [[0 + heuristic[start_point], path]]
    while len(frontier) > 0:
        curr_cost, curr_path = frontier.pop(0)
        n = curr_path[-1]
        curr_cost -= heuristic[n]
        if n in goals:
            return curr_path
        explored.append(n)
        children = [i for i in range(len(cost[0]))
            if cost[n][i] not in [0, -1]]
        for i in children:
            new_curr_path = curr_path + [i]
            new_path_cost = curr_cost + cost[n][i] + heuristic[i]
```

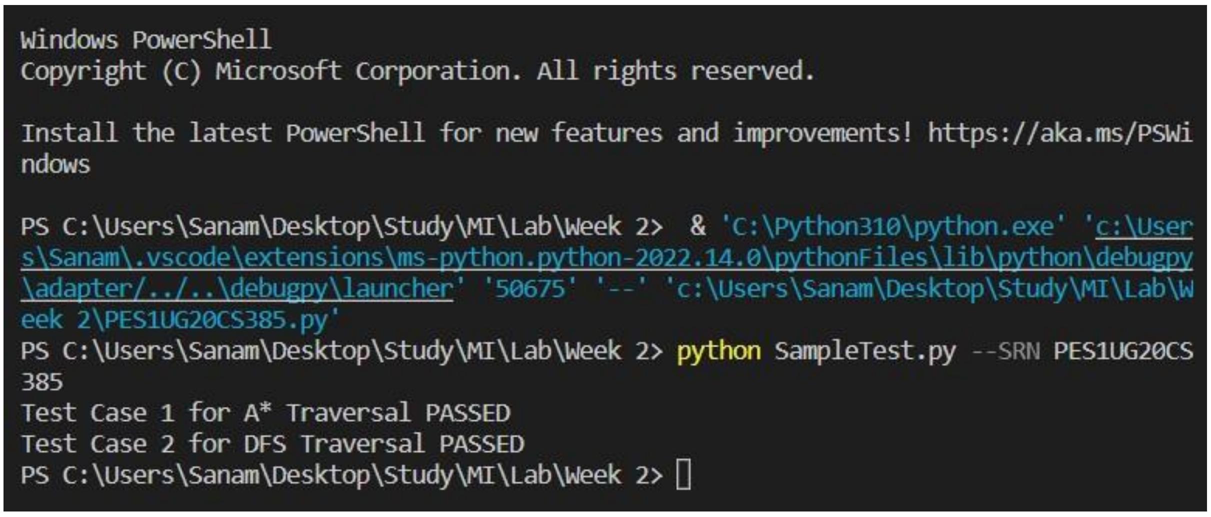


```
            if i not in explored and new_curr_path not in [i[1] for i in
frontier]:
                frontier.append((new_path_cost, new_curr_path))
                frontier = sorted(frontier, key=lambda x: (x[0], x[1]))
            elif new_curr_path in [i[1] for i in frontier]:
                index = search_q(frontier, new_curr_path)
                frontier[index][0] = min(frontier[index][0], new_path_cost)
                frontier = sorted(frontier, key=lambda x: (x[0], x[1]))
    return list()
visited = list(0 for i in range(0,11))
path = []



def DFS_Traversal(cost, start_point, goals):
    """
    Perform DFS Traversal and find the optimal path
        cost: cost matrix (list of floats/int)
        start_point: Staring node (int)
        goals: Goal states (list of ints)
    Returns:
        path: path to goal state obtained from DFS(list of ints)
    """
    path = []
    pathList = []
    visitedArray = [0 for _ in range(len(cost))]
    arrayAsStack = [(start_point, [start_point])]
    while (len(arrayAsStack) != 0):
        last, currPath = arrayAsStack[-1]
        if visitedArray[last] == 1:
            pass
        else:
            visitedArray[last] = 1
            if last in goals:
            # success, path found
                return currPath
            for element in range(len(cost) - 1, 0, -1):
                if cost[last][element] >= 1:
                    if visitedArray[element] == 1:
                        pass
                    else:
                        tempArray = [i for i in currPath]
                        tempArray.append(element)
                        arrayAsStack.append((element, tempArray))
    if len(path) == 0:
        return path
    else:
        return []
    return pat
```

OUTPUT :



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWi
ndows

PS C:\Users\Sanam\Desktop\Study\MI\Lab\Week 2>  & 'C:\Python310\python.exe' 'c:\User
s\Sanam\.vscode\extensions\ms-python.python-2022.14.0\pythonFiles\lib\python\debugpy
\adapter/../..\debugpy\launcher' '50675' '--' 'c:\Users\Sanam\Desktop\Study\MI\Lab\W
eek 2\PES1UG20CS385.py'
PS C:\Users\Sanam\Desktop\Study\MI\Lab\Week 2> python SampleTest.py --SRN PES1UG20CS
385
Test Case 1 for A* Traversal PASSED
Test Case 2 for DFS Traversal PASSED
PS C:\Users\Sanam\Desktop\Study\MI\Lab\Week 2> 
```