NAME: Hayden Cordeiro **ROLL NO: 05** **BATCH: D**

EXPERIMENT NO.: 1

Aim: To implement A* Algorithm

Learning Objective: To understand A*t and simulate the same in software.

Learning Outcome: Student are able to successfully simulate an A* algorithm.

Course Outcome:

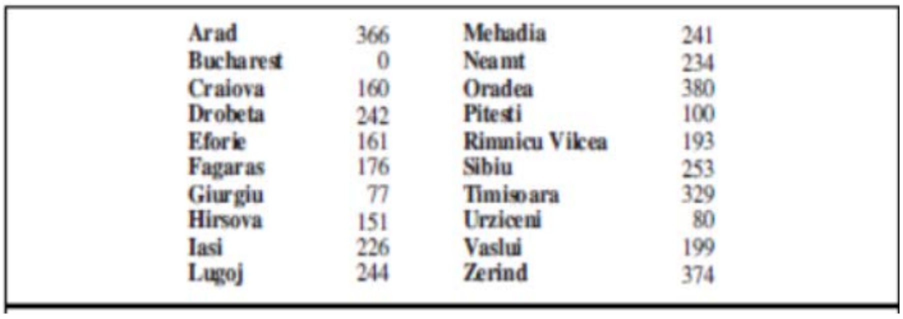
CSL703.1 To realize the basic techniques to build intelligent systems

Program Outcome:
(PO 3) Design/ development of solutions: Breadth and uniqueness of engineering problems i.e., the extent to
which problems are original and to which solutions have previously been identified or codified
(PO 12) Life Long Learning

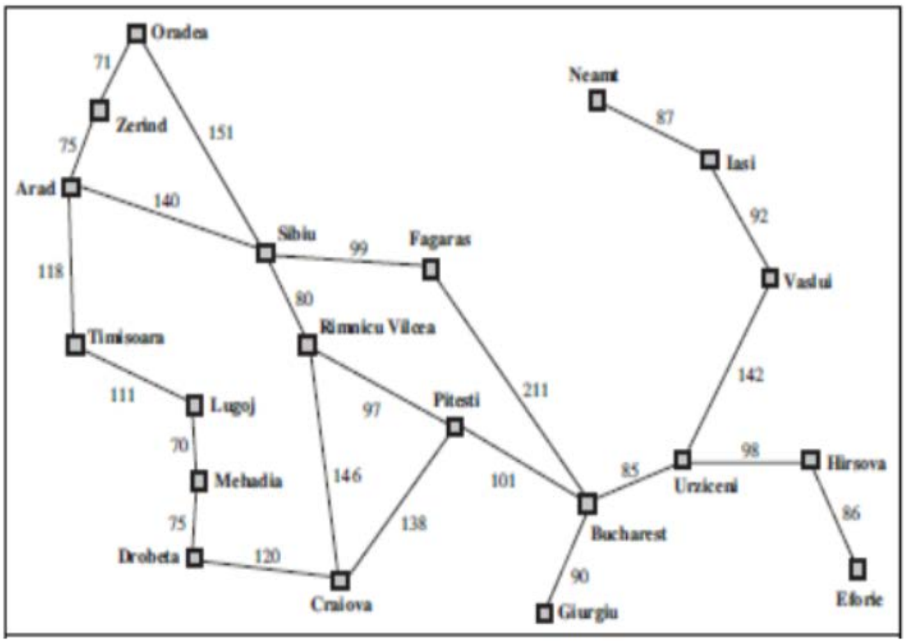
Bloom's Taxonomy Level:

- Remembering
- Understanding

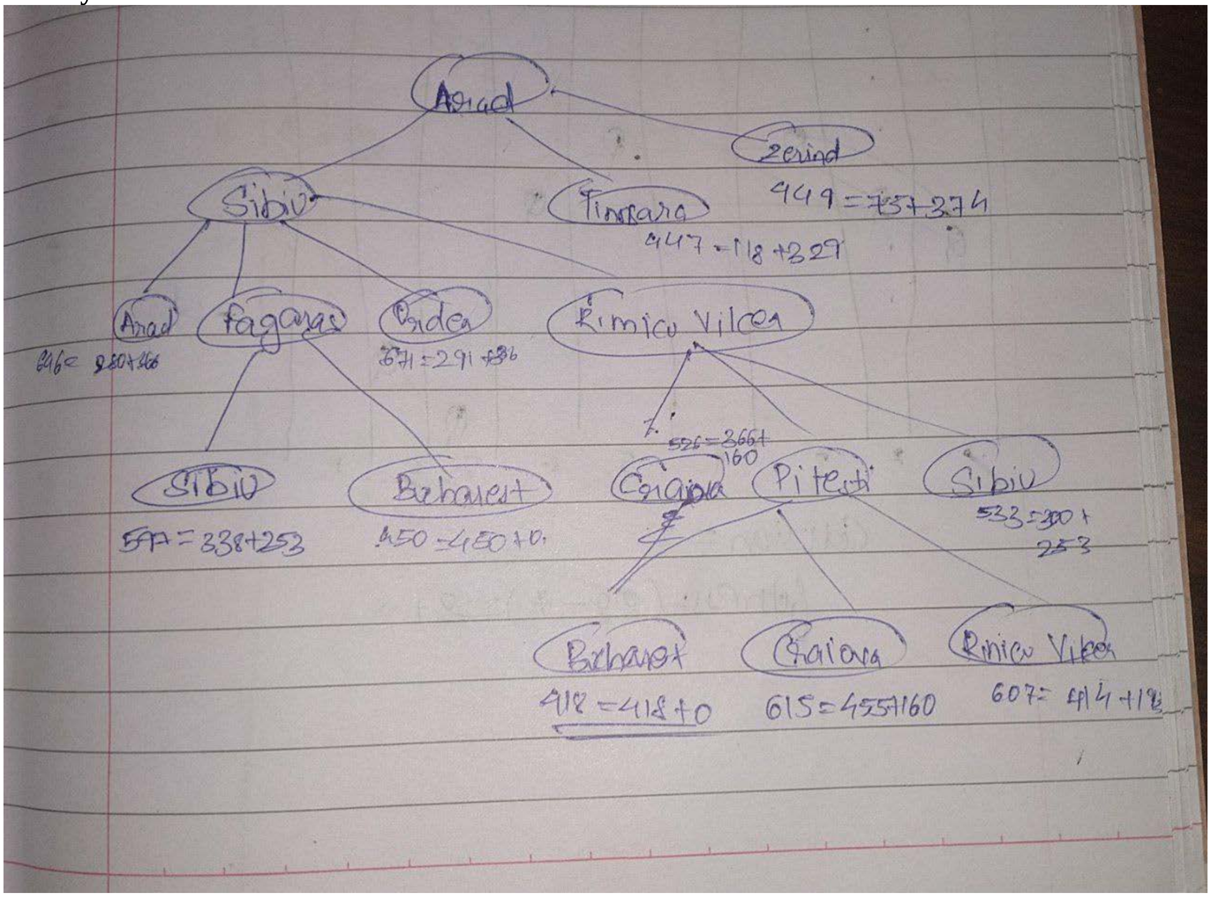
Input:



Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374



Theory:

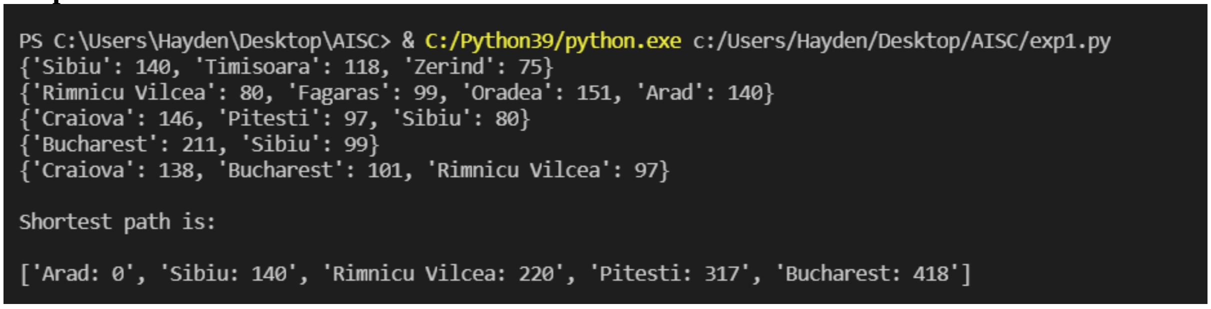


Algorithm:

```
1. Initialize the open list
2. Initialize the closed list
   put the starting node on the open
   list (you can leave its f at zero)

3. while the open list is not empty
   a) find the node with the least f on
      the open list, call it "q"

   b) pop q off the open list

   c) generate q's 8 successors and set their
      parents to q

   d) for each successor
       i) if successor is the goal, stop search
         successor.g = q.g + distance between
                          successor and q
         successor.h = distance from goal to
         successor (This can be done using many
         ways, we will discuss three heuristics-
```

```
          Manhattan, Diagonal and Euclidean
          Heuristics)

          successor.f = successor.g + successor.h

        ii) if a node with the same position as
            successor is in the OPEN list which has a
           lower f than successor, skip this successor

        iii) if a node with the same position as
            successor  is in the CLOSED list which has
            a lower f than successor, skip this successor
            otherwise, add  the node to the open list
     end (for loop)

    e) push q on the closed list
    end (while loop)
```

Output:



```
PS C:\Users\Hayden\Desktop\AISC> & C:/Python39/python.exe c:/Users/Hayden/Desktop/AISC/exp1.py
{'Sibiu': 140, 'Timisoara': 118, 'Zerind': 75}
{'Rimnicu Vilcea': 80, 'Fagaras': 99, 'Oradea': 151, 'Arad': 140}
{'Craiova': 146, 'Pitesti': 97, 'Sibiu': 80}
{'Bucharest': 211, 'Sibiu': 99}
{'Craiova': 138, 'Bucharest': 101, 'Rimnicu Vilcea': 97}

Shortest path is:

['Arad: 0', 'Sibiu: 140', 'Rimnicu Vilcea: 220', 'Pitesti: 317', 'Bucharest: 418']
```

Conclusion: The A* is successfully implemented