NAME: Meith Navlakha
SAP: 60004190068
AI Practical

EXPERIMENT 3

AIM: Implement A* Search Algorithm to reach the goal state.

THEORY:

A * algorithm is a searching algorithm that searches for the shortest path between the initial and the final state. It is used in various applications, such as maps. It is based on heuristic methods which helps to achieve optimality. It also offers completeness i.e., if there is any existing solution, then the algorithm will definitely find it. A* is a different form of the best-first algorithm.

When A* enters into a problem, firstly it calculates the cost to travel to the neighbouring nodes and chooses the node with the lowest cost. If The f(n) denotes the cost, A* chooses the node with the lowest f(n) value. Here 'n' denotes the neighbouring nodes. The calculation of the value can be done as shown below:

$$f(n)=g(n)+h(n)f(n)=g(n)+h(n)$$

where,
g(n) = shows the shortest path's value from the starting node to node n
h(n) = The heuristic approximation of the value of the node(Estimated Cost)

The heuristic value has an important role in the efficiency of the A* algorithm. To find the best solution, you might have to use different heuristic function according to the type of the problem.

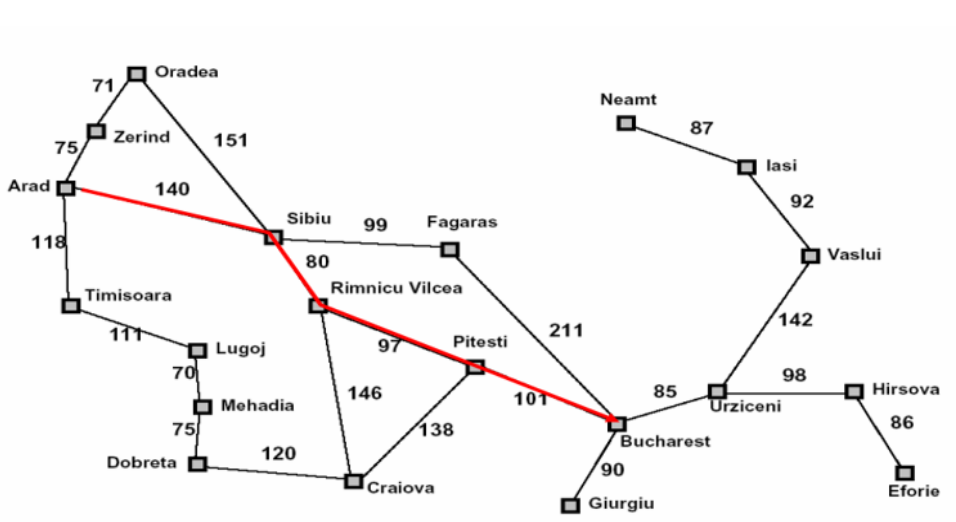
Problem:

- Creation of optimal heuristic functions is a difficult task.
- Space Complexity is $O(b^d)$, as it stores all generated nodes in memory.

Problem Statement: Identify and analyze informed search Algorithm to find an optimal solution from Arad to Bucharest on the given map of Romania. Implement A* Algorithm to reach to the goal state.

NAME: Meith Navlakha
SAP: 60004190068
AI Practical

Map of Romania



$h(n)$

Straight–line distance to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

CODE:

```
import java.util.*;
import java.util.Arrays;
import java.lang.Integer;
public class AStar {
  public static void calcAstar(int[][] gN, int[] hN, String[] city, int cur, int goal, int N) {
    int min = Integer.MAX_VALUE;
    int minIndex = 0;
    int[] visited = new int[N];
    visited[cur] = 1;

    System.out.print("\n\n ***** A* PATH *****");
    System.out.print("\n\n " + city[cur] + " ");

    while (cur != goal) {

      for (int j = 0; j < N; j++) {

        if (gN[cur][j] > 0) {

          if (visited[j] != 1 && min > gN[cur][j] + hN[j]) {
            min = gN[cur][j] + hN[j];
            minIndex = j;
          }
        }
      }
```

```
      System.out.print(" -> " + city[minIndex] + " ");
      cur = minIndex;
      min = Integer.MAX_VALUE;
      visited[minIndex] = 1;

    }

  }

  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Enter the number of Nodes : ");
    // int n = sc.nextInt() ;
    int N = 20;

    // int[] hN = new int[N] ;
    // int[] city = new int[N] ;
    int[][] gN = new int[N][N];
int[] hN = {366,0,160,242,161,178,77,151,226,244,241,234,380,98,193,253,329,80,199,374
} ;
String[] city = {"Arad" , "Bucharest" , "Craivoa" , "Dobreta" , "Eforie","Fagaras", "Guirgiu"
,"Hirsova", "Iasi", "Lugoj" ,"Mehadia" , "Neamt" , "Oradea", "Pitesti","Rimnieu Vilcea" ,
"Sibiu", "Timisoara", "Urzieeni","Vaslui" , "Zerind" } ;
    System.out.print("\n\n HEURISTIC VALUES : ");
    for (int i = 0; i < N; i++) {
      if (i == 14) {
        System.out.print("\n " + city[i] + ": " + hN[i]);

      } else {
        System.out.print("\n " + city[i] + " : \t " + hN[i]);

      }

    }

    // Distances
    System.out.print("\n\n G(n) for CITIES \n");

    for (int i = 0; i < N; i++) {
      for (int j = 0; j < N; j++) {
        gN[i][j] = sc.nextInt();
      }
}
    calcAstar(gN, hN, city, 0, 1, N);
  }
}
```

NAME: Meith Navlakha
SAP: 60004190068
AI Practical

OUTPUT

```
HEURISTIC VALUES :
Arad :              366
Bucharest :         0
Craivoa :           160
Dobreta :           242
Eforie :            161
Fagaras :           178
Guirgiu :           77
Hirsova :           151
Iasi :              226
Lugoj :             244
Mehadia :           241
Neamt :             234
Oradea :            380
Pitesti :           98
Rimnieu Vilcea:     193
Sibiu :             253
Timisoara :         329
Urzieeni :          80
Vaslui :            199
Zerind :            374

***** A* PATH *****

 Arad  -> Sibiu  -> Rimnieu Vilcea  -> Pitesti  -> Bucharest
C:\Users\meith\Desktop\SEM 5\AI>
```

CONCLUSION: In this experiment I implemented A* Algorithm to identify and analyze informed search Algorithm to find an optimal solution from Arad to Bucharest on the given map of Romania. If the algorithm had only considered h(n), based on the incurring cost then, Fagaras (h(n)= 178) would have been selected instead of Rimnieu Vilcea(h(n)=193) thus deviating from the optimal path. But when considering f(n) = g(n)+h(n), that is considering the actual cost of traveling between the nodes along with the incurring cost, then Rimnieu Vilcea is selected over Fagaras. Thus, A* Algorithm gives an optimal, complete solution.