ARTIFICIAL INTELLIGENCE

AI Search Strategies

UEL Module Code **EG7424**	*UEL Module Name* **Artificial Intelligence**	
ASU Course Code **CSE472**	*ASU Course Name* **Artificial Intelligence**	
	Semester **Spring 2022**	*Due Date* **22 /05 /2022**

Submitted to

Dr. Ammar Mohamed

Eng. Mahmoud Soheil

Submitted by

Name	ASU ID	UEL ID
Mahmoud Mohamed Seddik Elnashar	19P3374	U2140584
Zakaria Sobhy Abd El-Salam Soliman	19P2676	U2140654
Abdelrahman Mohamed Salah Eldin	19P9131	U2140523
Mohamed Aly Elsayed Matar	19P5238	U2140601
Omar Ashraf Mabrouk	19P8102	U2140624

Table of Contents

User Manual

0. Running Executable

Run the Executable Team4.exe to start the program.

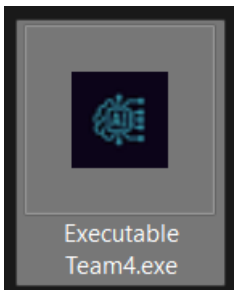

Figure 1 Shows exe

1. Startup Window

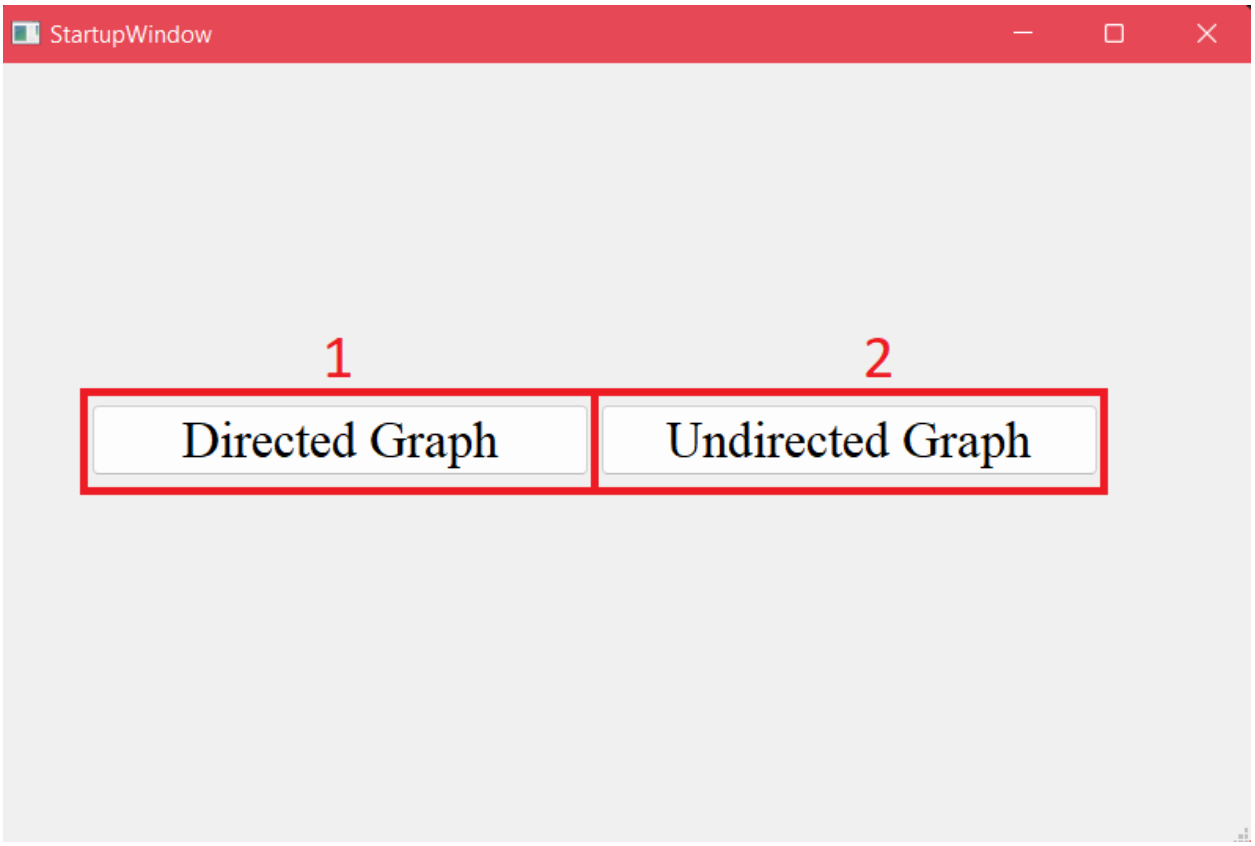

Figure 2 Shows Startup Window

This is the first window that appears to the user upon startup of the application.

In this window the user has the option to choose one of the two types of graphs:

1. Directed Graph: where the relationship between the nodes is a one-way relationship through the edges.
2. Undirected Graph: where the relationship between the nodes is a two-way relationship through the edges.

After choosing the type of graph a transition occurs to another window (Main window).

2.0. Main Window

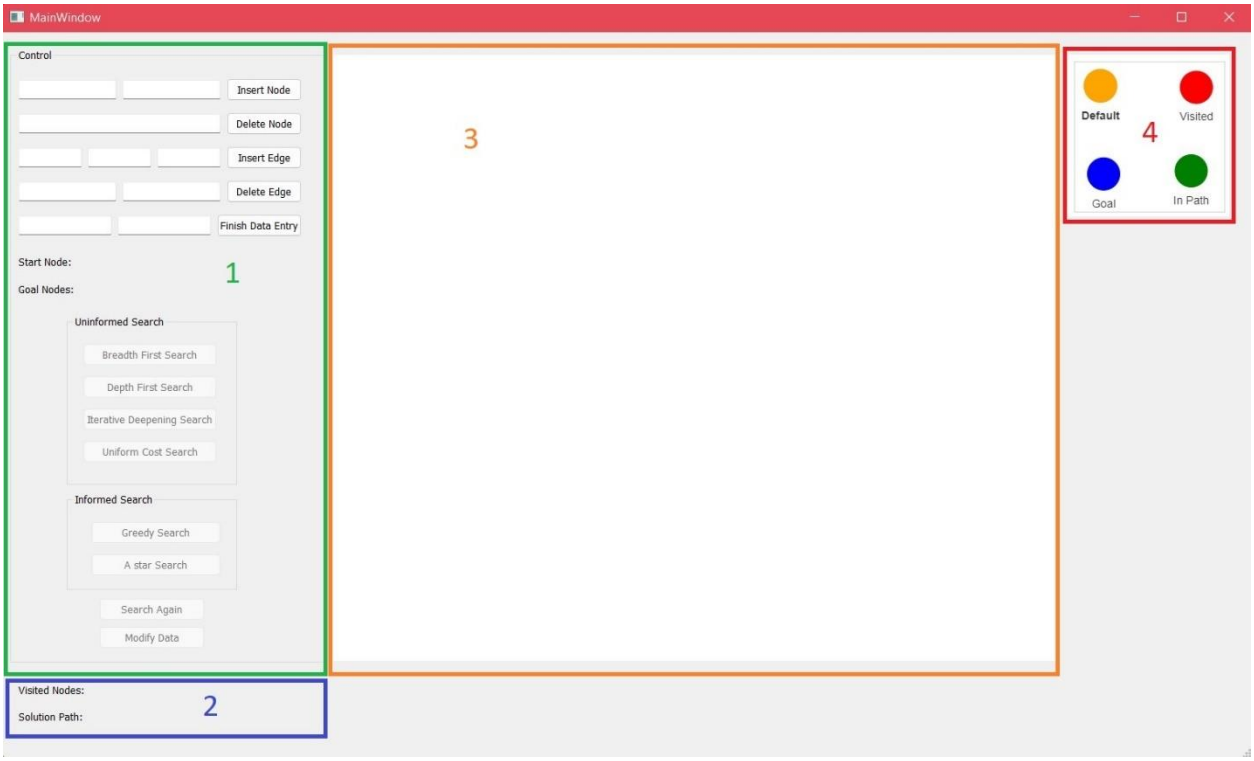

Figure 3 Shows Main Window

The Main Window consists of four main components:

1. Control panel: used to enter the graph data and choose the searching algorithm
2. Output panel: displays the visited nodes and the solution path which the chosen algorithm returns.
3. Graph view: shows visualization of the entered graph containing the nodes and the edges.
4. Legend: helps the user understand the color scheme of the graph.

2.1. Control Panel

2.1.1. Data entry

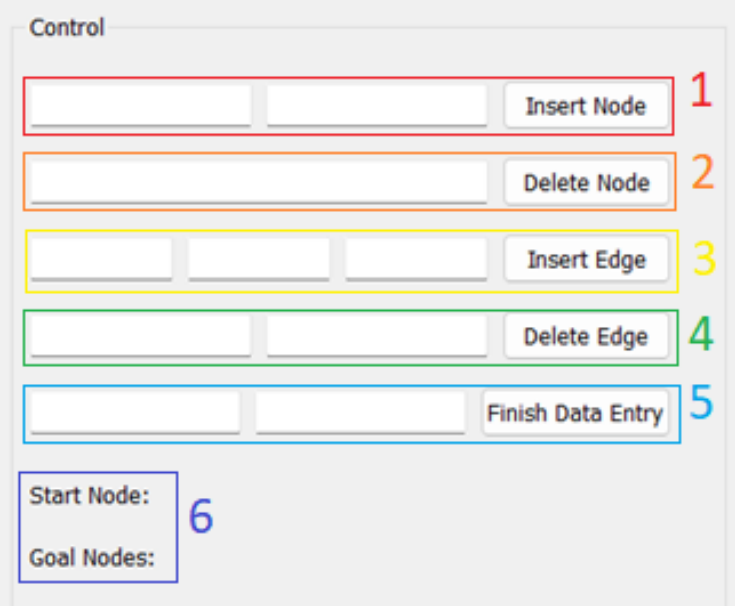

Figure 4 Shows Control Panel

1. Insert Node: enter the node name in the first textbox and the heuristic in the second text box then press Insert Node. The node will be displayed in the graph view. (Note: the node name is unique, the heuristic must be positive or zero).
2. Delete Node: enter the node name to be deleted and press Delete Node. The node will be removed from the graph view.
3. Insert Edge: enter the first node name in the first textbox, the second node name in the second textbox and the weight in the third text box then press Insert Edge. The edge will be displayed in the graph view connecting the two nodes. (Note: in case of directed graph the edge is directed first → second in case of undirected order doesn't matter).
4. Delete Edge: enter the first node in the first text box and the second node in the second text box then press Delete Edge. The edge will be removed from the graph view.
5. Finish Data Entry: enter the start node in the first textbox and the goal nodes separated by commas in the second textbox then press Finish Data Entry. The color of the goal nodes will be changed in the graph view from orange (default) to blue (goal nodes).
6. Upon clicking Finish Data Entry the start and goal nodes are written here for display.

Example:

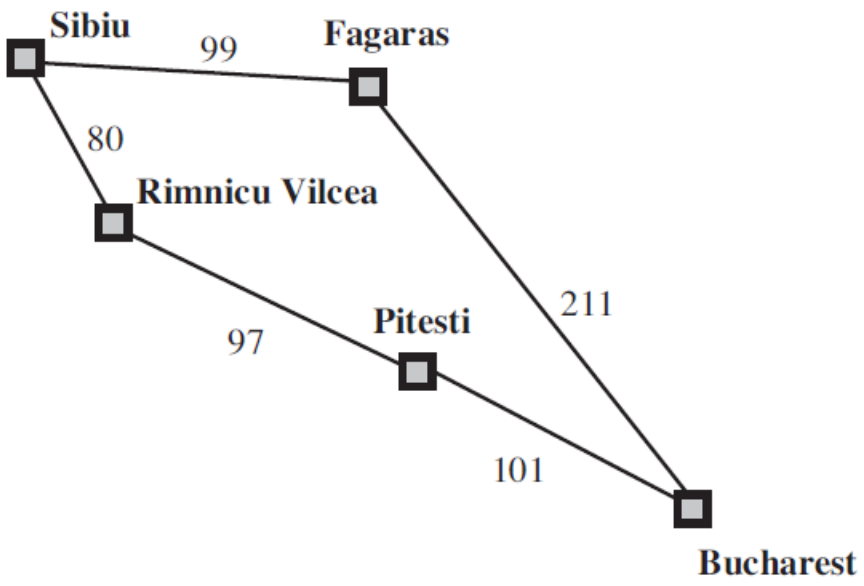

Figure 5 Example to be used throughout the manual

Step 1: Inserting nodes: type the node name and its heuristic then press Insert Node

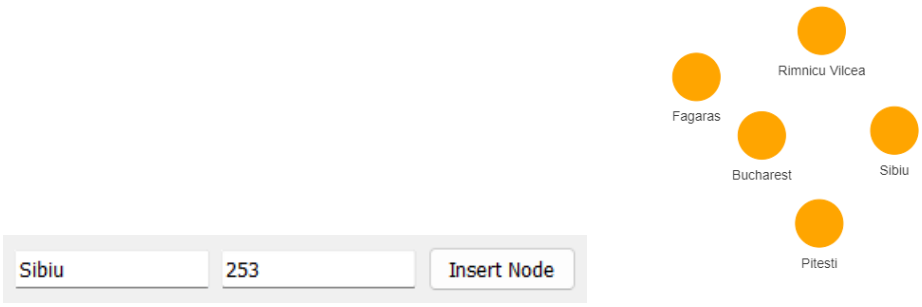

Figure 6 inserting nodes syntax

Deleting node: type the node name to be deleted then press Delete Node

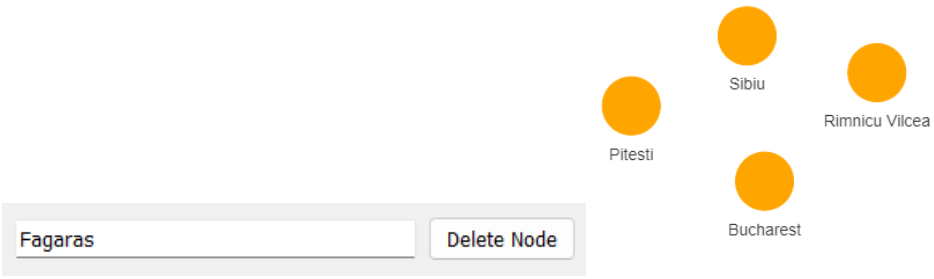

Figure 7 Deleting Nodes syntax

Step 2: Inserting edges: type first node, second node and the weight then press Insert Edge

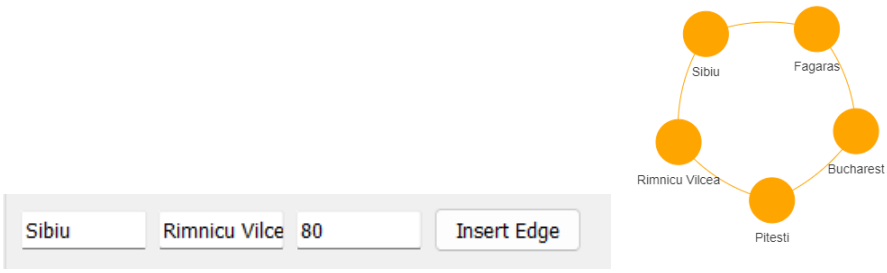

Figure 8 Inserting Edges syntax

Deleting edge: type the two node's name connected by an edge then press Delete Edge.

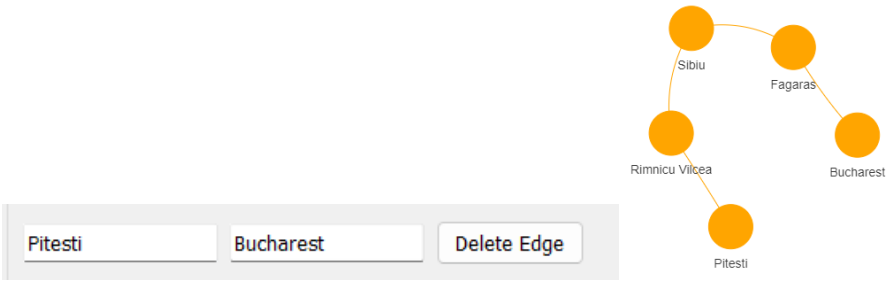

Figure 9 Deleting Edges syntax

Step 3: Finish Data Entry: type the start and then the goal nodes separated by commas (',') then press Finish Data Entry.

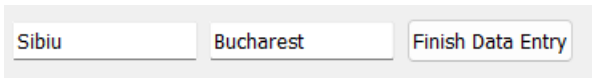

Figure 10 Start and Goal nodes syntax

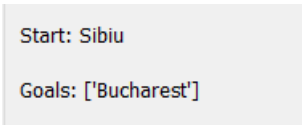

Figure 11 Start and Goal nodes displayed

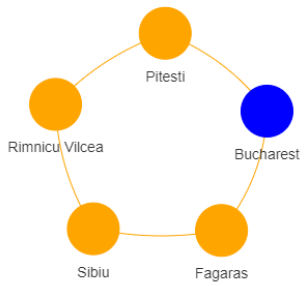

Figure 12 Graph after inserting all data

2.1.2. Choosing searching algorithm

The user has a variety of options in choosing the type of searching algorithms either uninformed search or informed search.

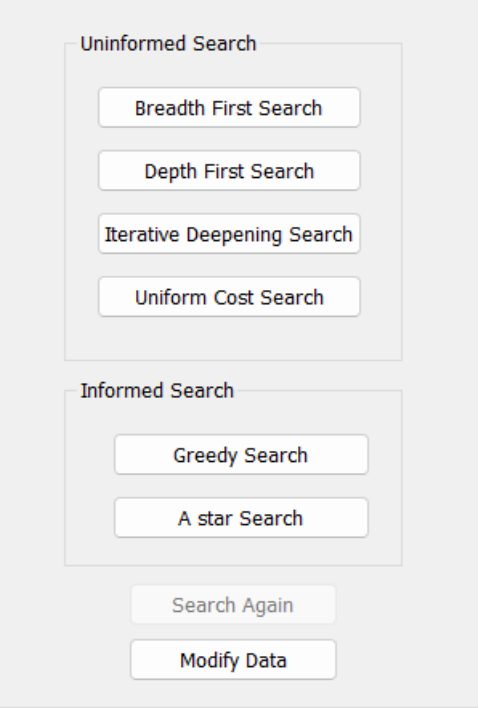

Figure 13 Shows searching algorithms available

Uninformed Search include:

Breadth first, Depth first, Iterative deepening and uniform cost.

Informed Search include:

Greedy and A*.

1. Searching:

   Simply the user chooses whichever searching algorithm he wishes to use.

   If the user wants to search on the same graph using another algorithm, he first presses the Search Again button and then choose the algorithm he wishes to execute.

2. Modifying Data:

   If the user wants to modify the data i.e., add/delete nodes/edges he presses the Modify Data button then he can modify the data.

2.2. Output panel

This panel shows the list of visited nodes in order and the solution path.

```
Visited Nodes: ['Sibiu', 'Rimnicu Vilcea', 'Fagaras', 'Pitesti', 'Bucharest']
Solution Path: ['Sibiu', 'Rimnicu Vilcea', 'Pitesti', 'Bucharest']
```

Figure 14 Shows the visited nodes and the solution path using uniform cost search on the previous example

2.3. Graph View

Shows visualization of the entered graph containing the nodes and the edges.

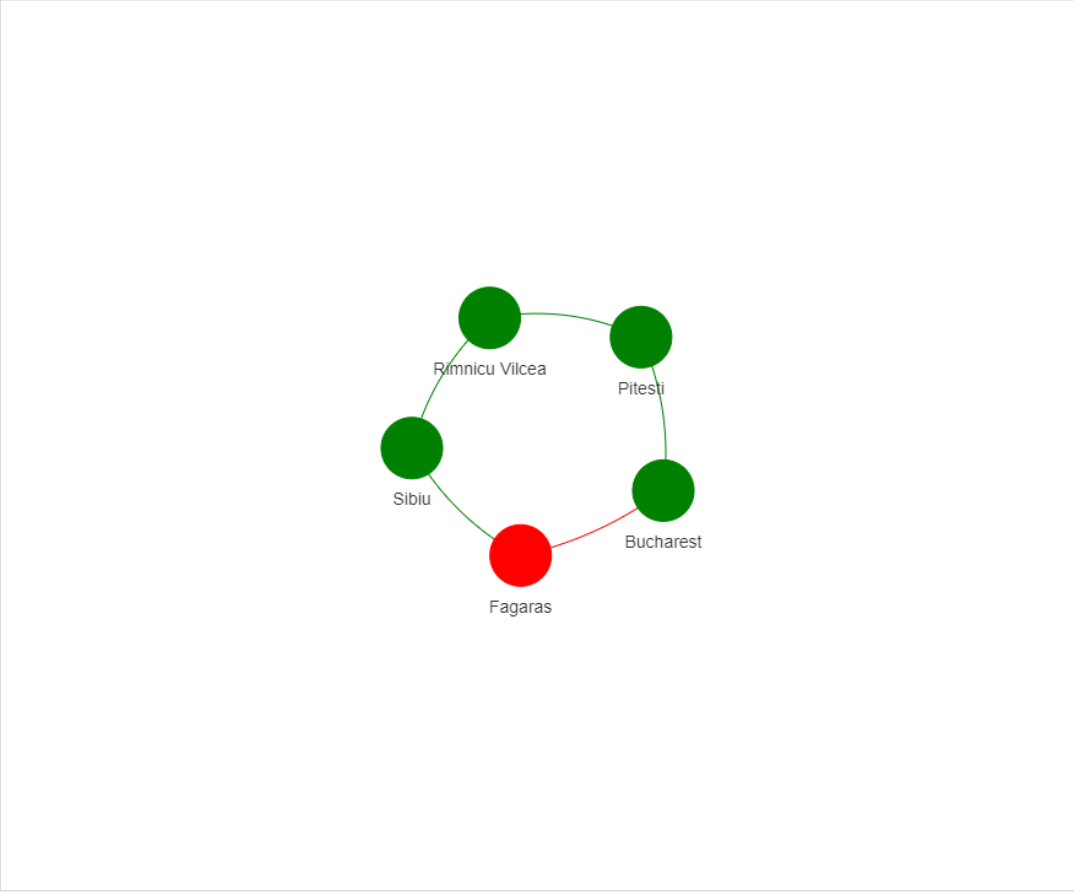

Figure 15 Graph after running uniform cost search on the previous Example

2.4. Legend

Helps the user understand the color scheme of the graph.

- The orange color is used as a default before executing search and the color of unvisited nodes.
- The blue color is for the goal node after being identified.
- The red color is for the visited nodes after executing a search.
- The green color is for the nodes in the solution path after executing a search.

Figure 16 Shows the legend

3.0. Test Example

The goal is to get from Arad to Bucharest we will test it using all the searching algorithms.

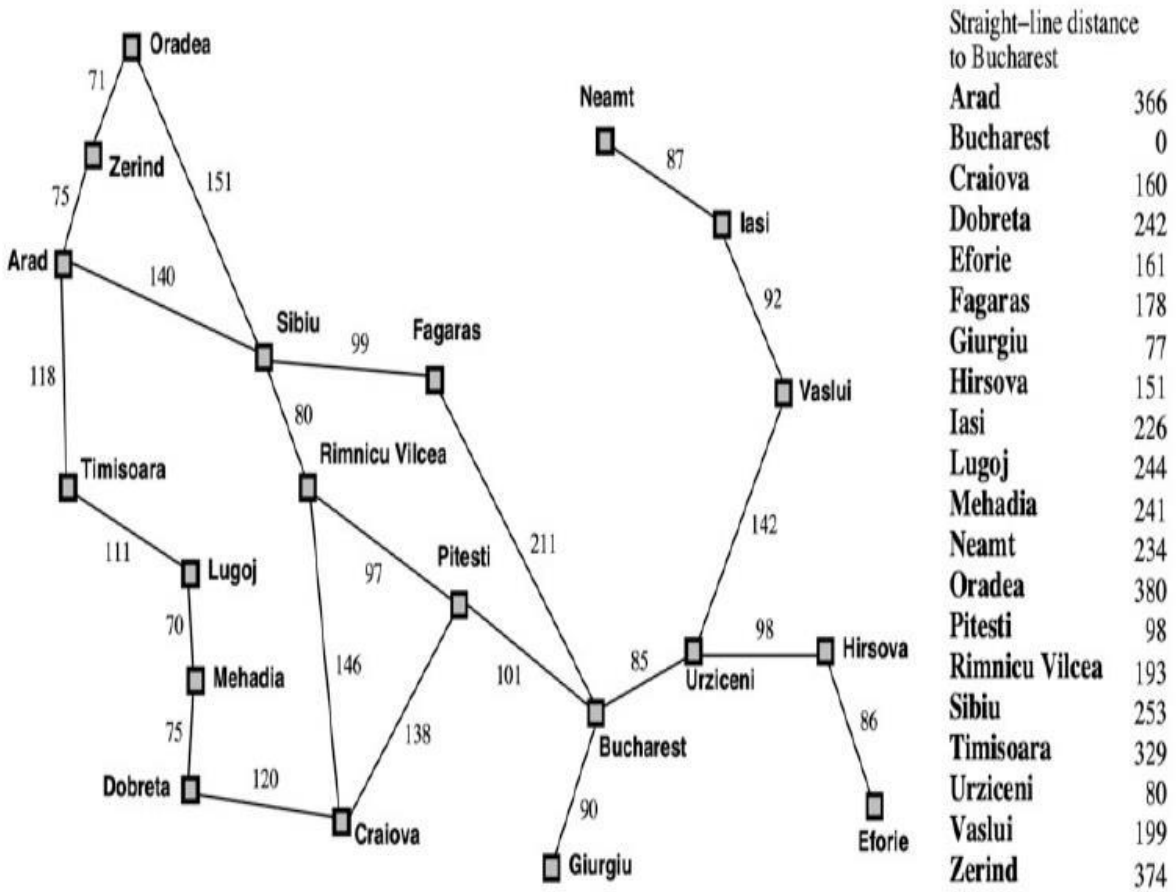

Figure 17 Shows Example from sheet 3

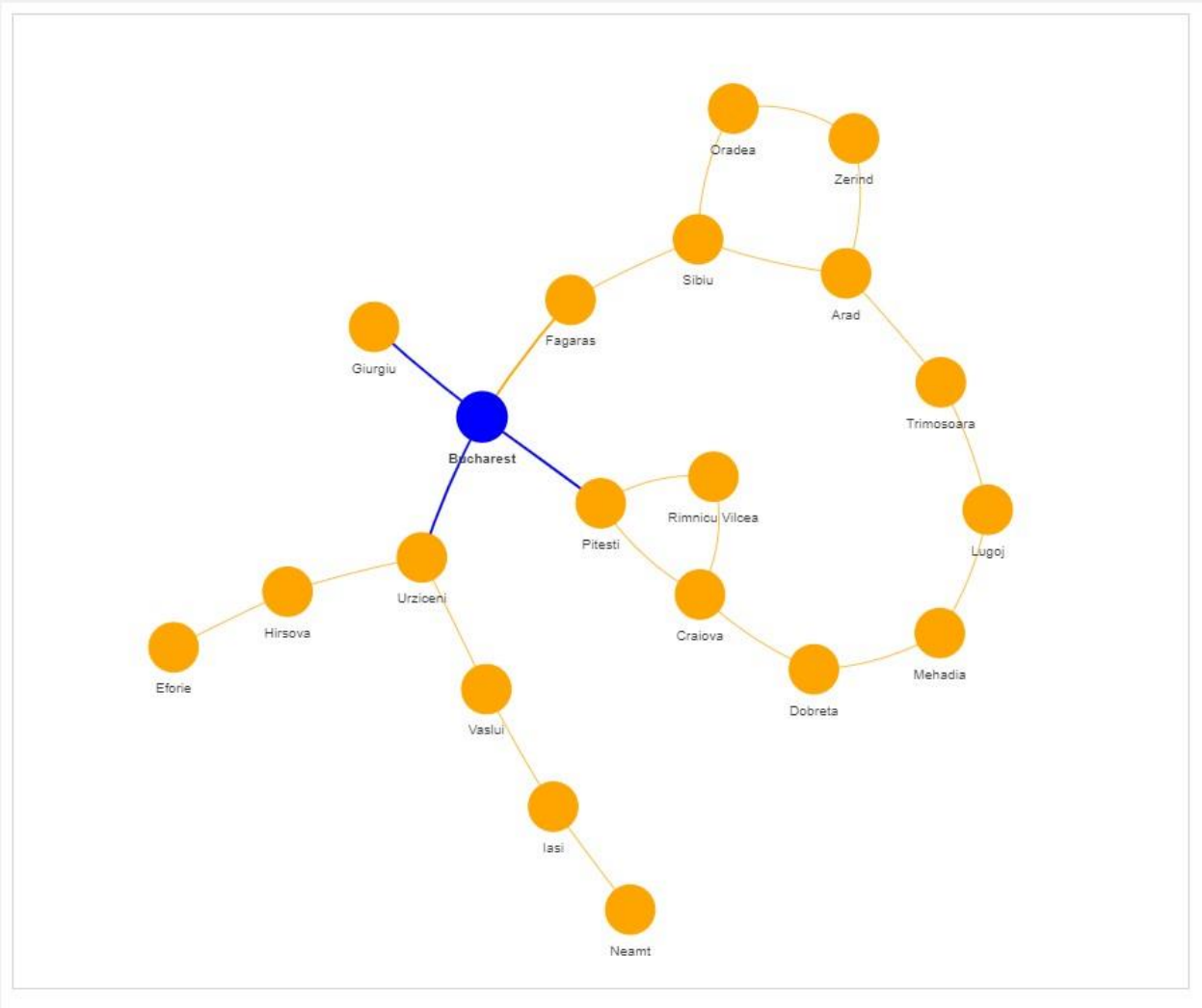

Figure 18 Initial state with Bucharest as goal

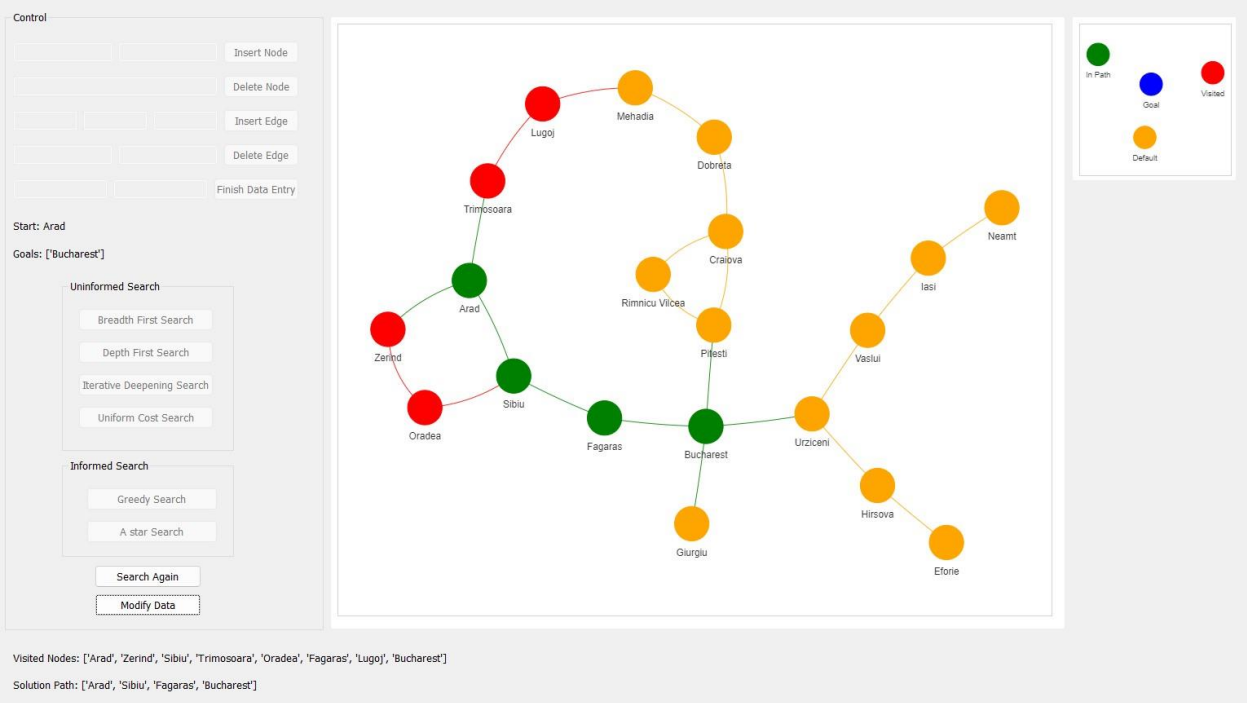

Figure 19 Breadth First Search

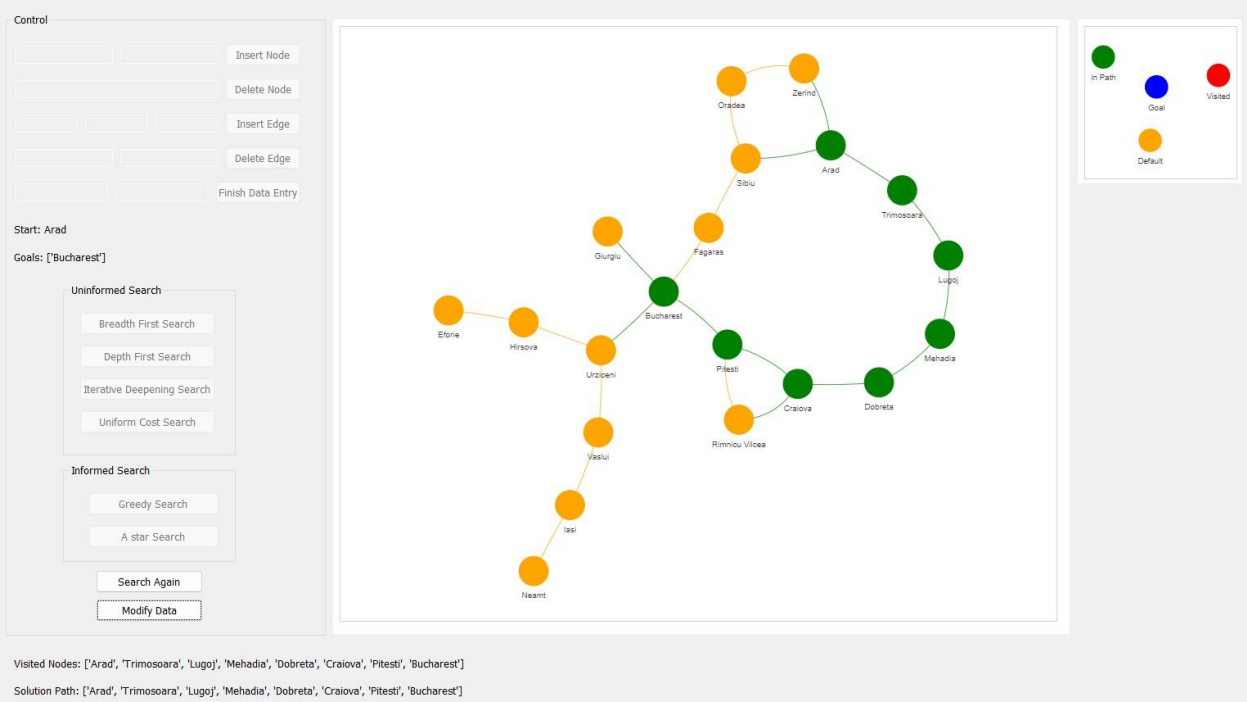

Figure 20 Depth First Search

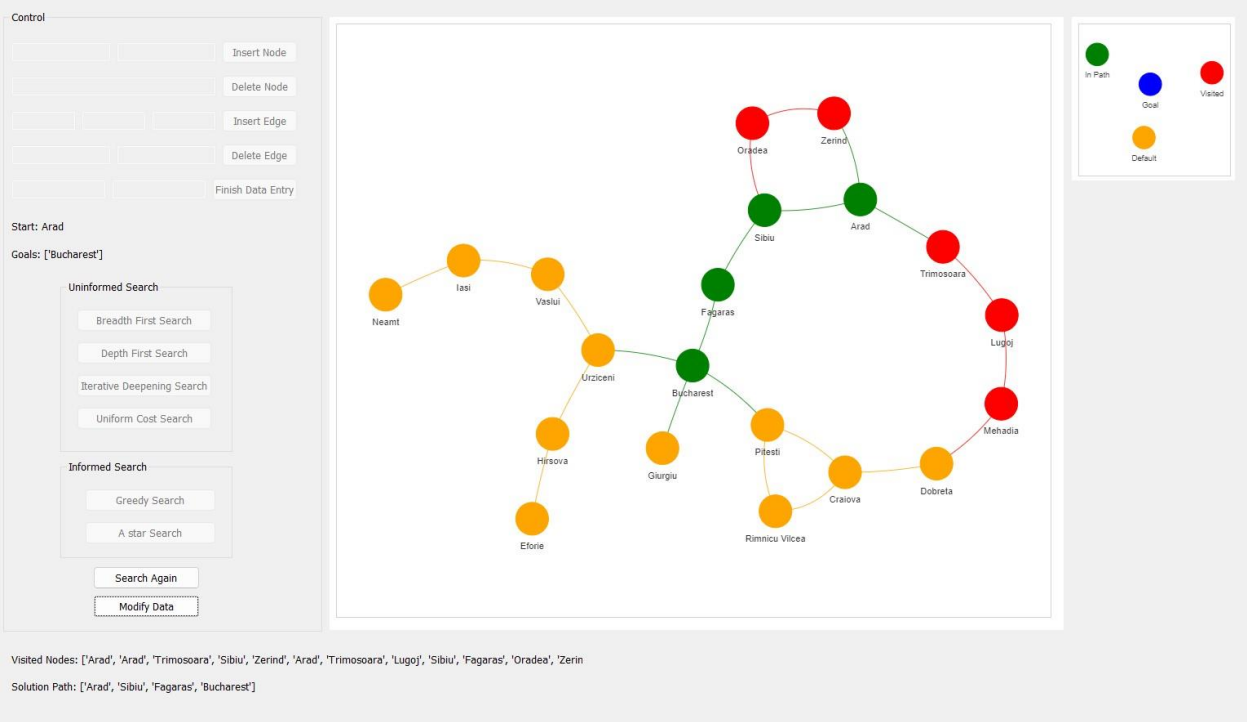

Figure 21 Iterative Deepening Search

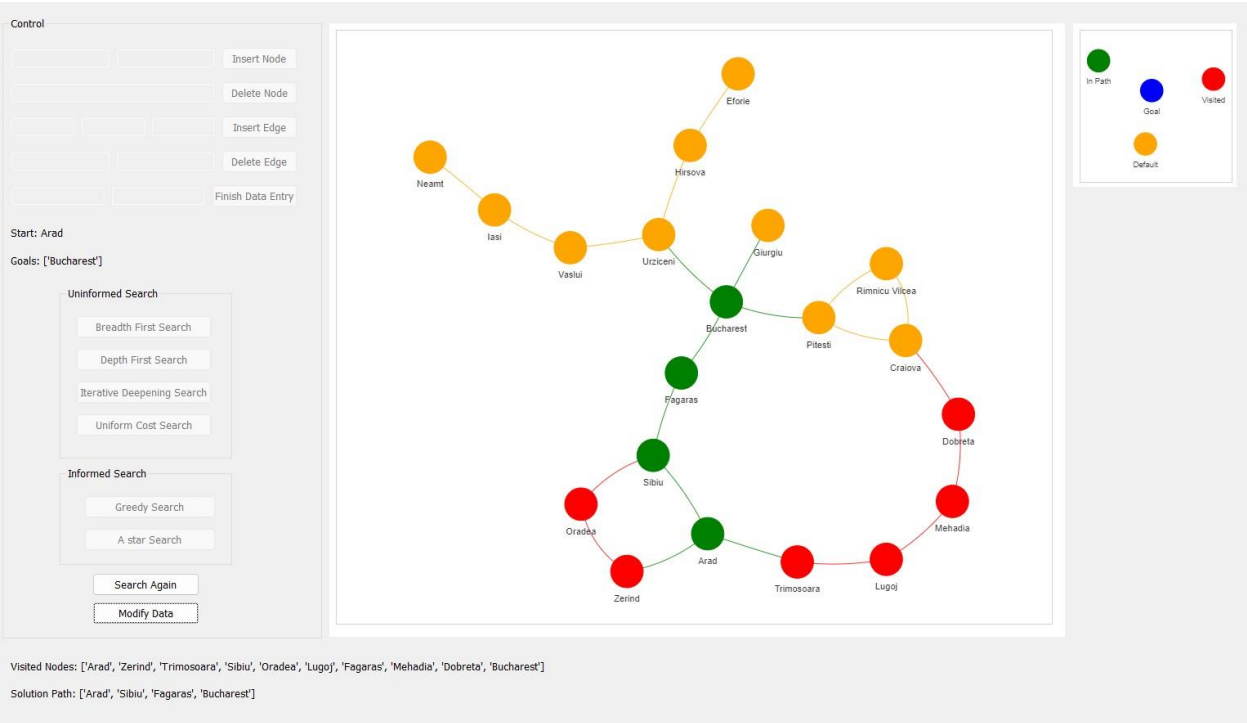

Figure 22 Uniform Cost Search

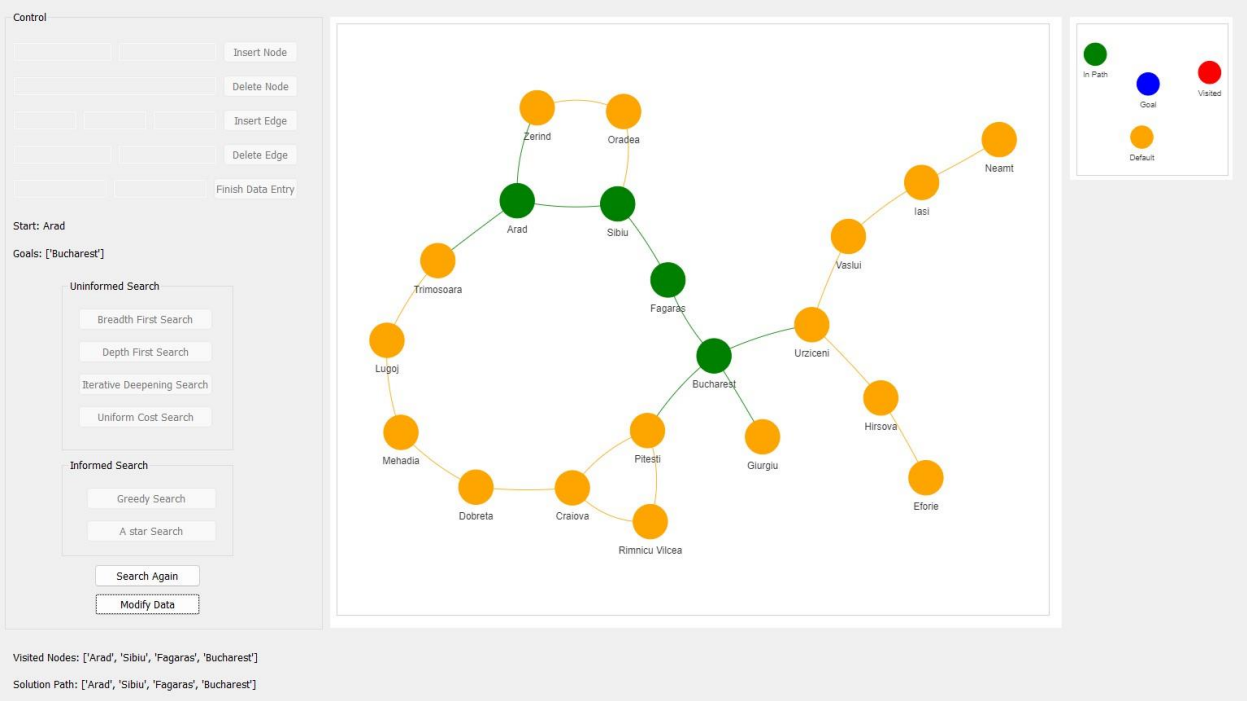

Figure 23 Greedy Search

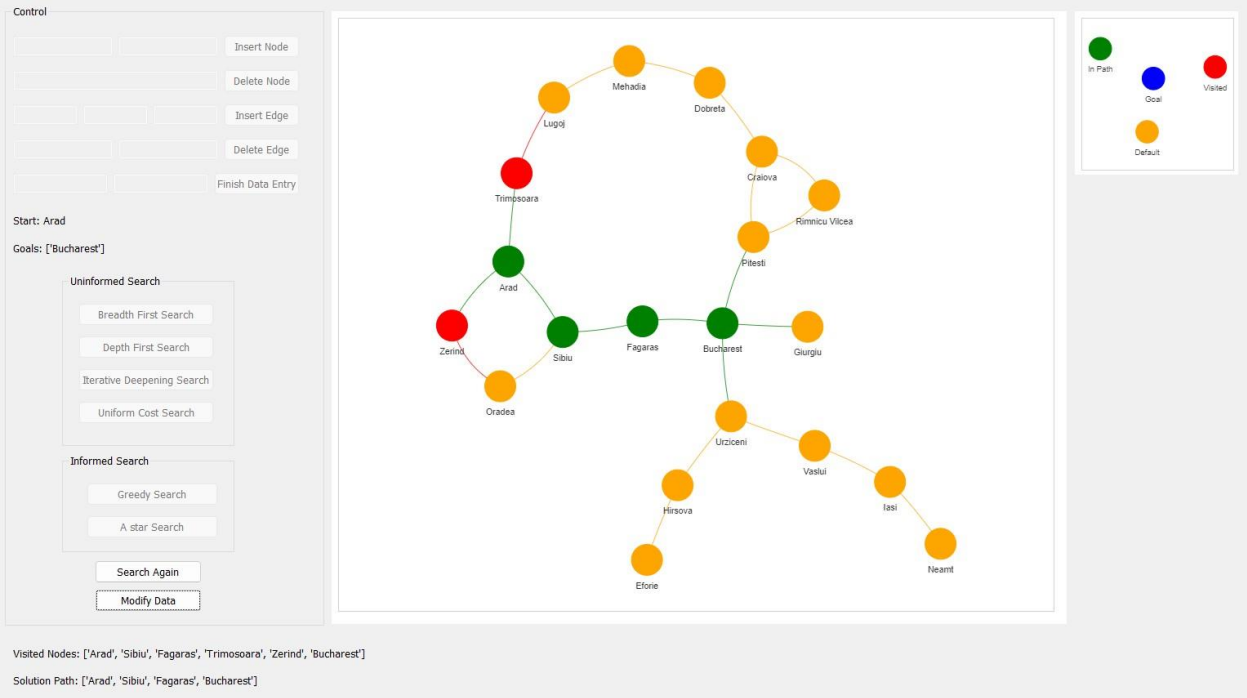

Figure 24 A Search*