# Romania Problem Report

## Procedure to Compile and Run

The Romania problem was solved using a graph representation of the Romania map. User will be able to search the shortest path from a define source to the destination (Bucharest)using three search strategies such as,

- Breadth First Search (BFS)
- Depth First Search (DFS) and
- Iterative-Deepening Search (IDS).

The solution was programmed using Java programming language. The program includes three java files as follows.

1. **ShotestPathFinder.**java – the main program
2. **RoadMapGraph.java** – class that represents the Romania map as a graph and its functionalities.
3. **City.java** – class represents a vertex in the Romania Map represented by a graph.

To compile and run the program the main method in the java file named "ShotestPathFinder.java" should be executed using any IDE that supports Java. The program will provide a non-GUI representation of the graph in following forms.

- Adjacency list
- Adjacency Matrix

Program will ask the user for two inputs such as the search algorithm and the starting city. The answers need to be provided as instructed.

- For the search Algorithm user must enter 1 for BFS, 2 for DFS and 3 for IDS
- For the starting city user should provide the first letter of the name of the required city
  Ex: If the starting city is Arad, then user should input A or a.

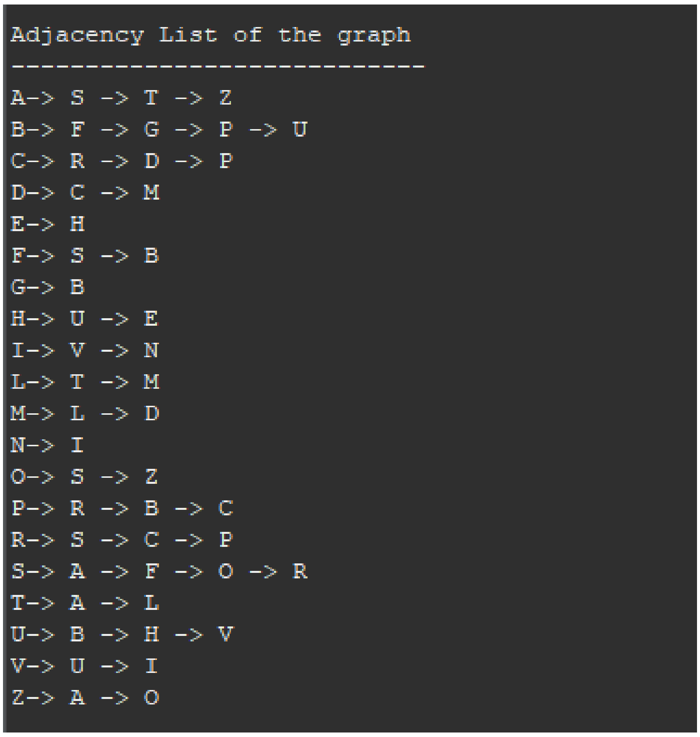
Once the inputs are given correctly, the program will present the following results,

- List of Explored/Visited cities.
- Path from starting city to destination city
- Path cost
- Iteration information for IDS.

# Results

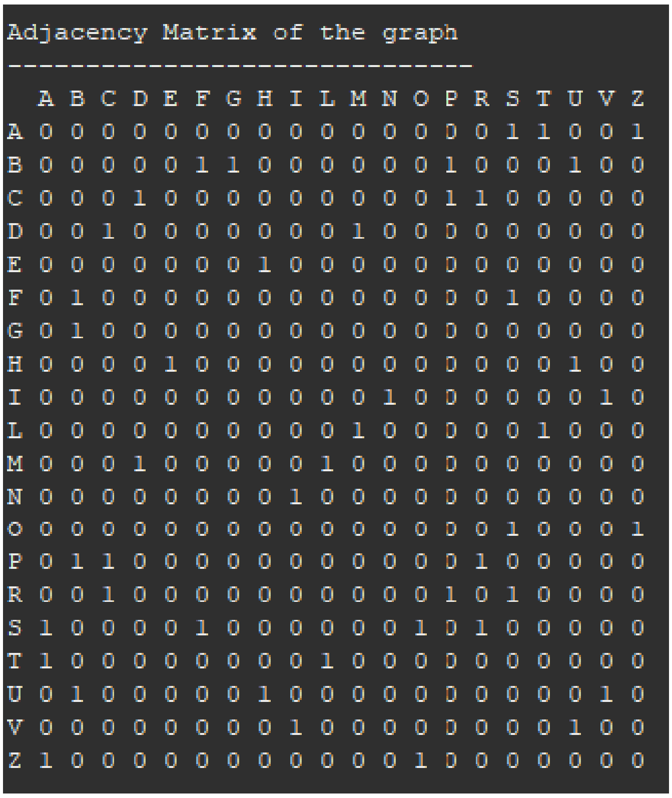
## Part 1: Non-GUI Representation of the Romania map

1. Adjacency List



```
Adjacency List of the graph
----------------------------
A-> S -> T -> Z
B-> F -> G -> P -> U
C-> R -> D -> P
D-> C -> M
E-> H
F-> S -> B
G-> B
H-> U -> E
I-> V -> N
L-> T -> M
M-> L -> D
N-> I
O-> S -> Z
P-> R -> B -> C
R-> S -> C -> P
S-> A -> F -> O -> R
T-> A -> L
U-> B -> H -> V
V-> U -> I
Z-> A -> O
```

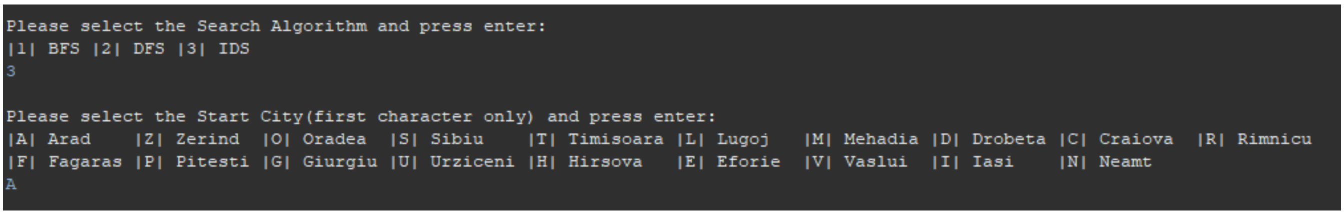
2. Adjacency Matrix



```
Adjacency Matrix of the graph
--------------------------------
  A B C D E F G H I L M N O P R S T U V Z
A 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1
B 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 1 0 0
C 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
D 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
E 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
F 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
G 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
H 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
I 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0
L 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0
M 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
N 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
O 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
P 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
R 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0
S 1 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0
T 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
U 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0
V 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0
Z 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
```

## Part2: Searching the Shortest Path

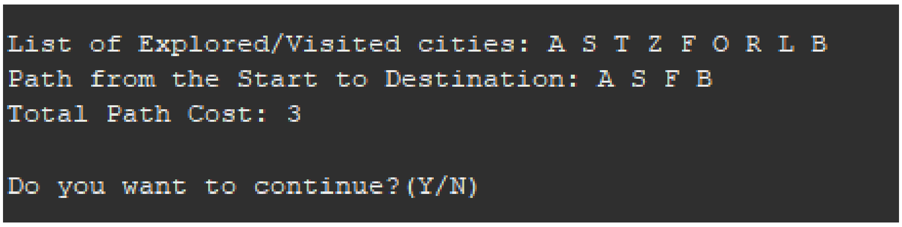
### User Inputs



```
Please select the Search Algorithm and press enter:
|1| BFS |2| DFS |3| IDS
3

Please select the Start City(first character only) and press enter:
|A| Arad    |Z| Zerind  |O| Oradea  |S| Sibiu    |T| Timisoara |L| Lugoj   |M| Mehadia |D| Drobeta |C| Craiova  |R| Rimnicu
|F| Fagaras |P| Pitesti |G| Giurgiu |U| Urziceni |H| Hirsova   |E| Eforie  |V| Vaslui  |I| Iasi    |N| Neamt
A
```

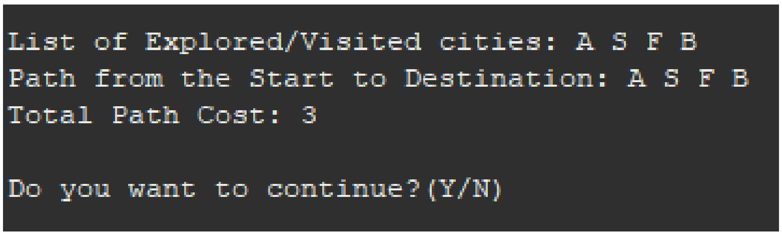
### Results for BFS (Starting city: Arad)



```
List of Explored/Visited cities: A S T Z F O R L B
Path from the Start to Destination: A S F B
Total Path Cost: 3

Do you want to continue?(Y/N)
```

### Results for DFS (Starting city: Arad)

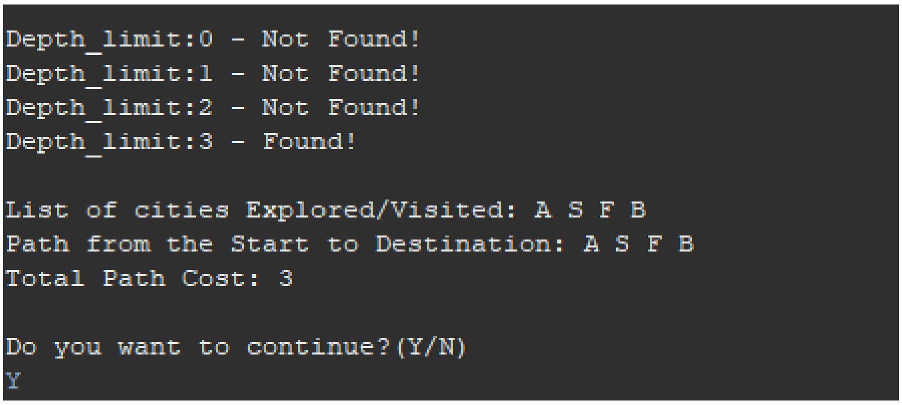


```
List of Explored/Visited cities: A S F B
Path from the Start to Destination: A S F B
Total Path Cost: 3

Do you want to continue?(Y/N)
```

### Results for IDS (Starting city: Arad)

This shows the list of explored nodes in the last iteration.



```
Depth_limit:0 - Not Found!
Depth_limit:1 - Not Found!
Depth_limit:2 - Not Found!
Depth_limit:3 - Found!

List of cities Explored/Visited: A S F B
Path from the Start to Destination: A S F B
Total Path Cost: 3

Do you want to continue?(Y/N)
Y
```