

# AI Project Report

---

**Aim** - Create a GUI, with the map of Romania as input. Apply Iterative depth first search algorithm on the Romania map and show the solution in each step. Give a provision to select any source and destination by the user. On clicking the “Submit” button, algorithm execution will be visualised on the screen.

## **Team Members -**

A-25 Abhash Goyal

A-31 Adwait Putambekar

A-32 Aman Joharapurkar

A-35 Anuj Belgaonkar

A-39 Atharva Telrande

## **Programming Language -**

Python

## **Libraries -**

time

pygame

math

tkinter

random

## **Functionality:**

- The program allows the user to input the type of traversal( supports only IDDFS currently), source city and destination city.
- It visualises the traversal process on a map of Romania using Pygame.
- It displays the final path from the source to the destination using a message box.

Romania Map Trave...

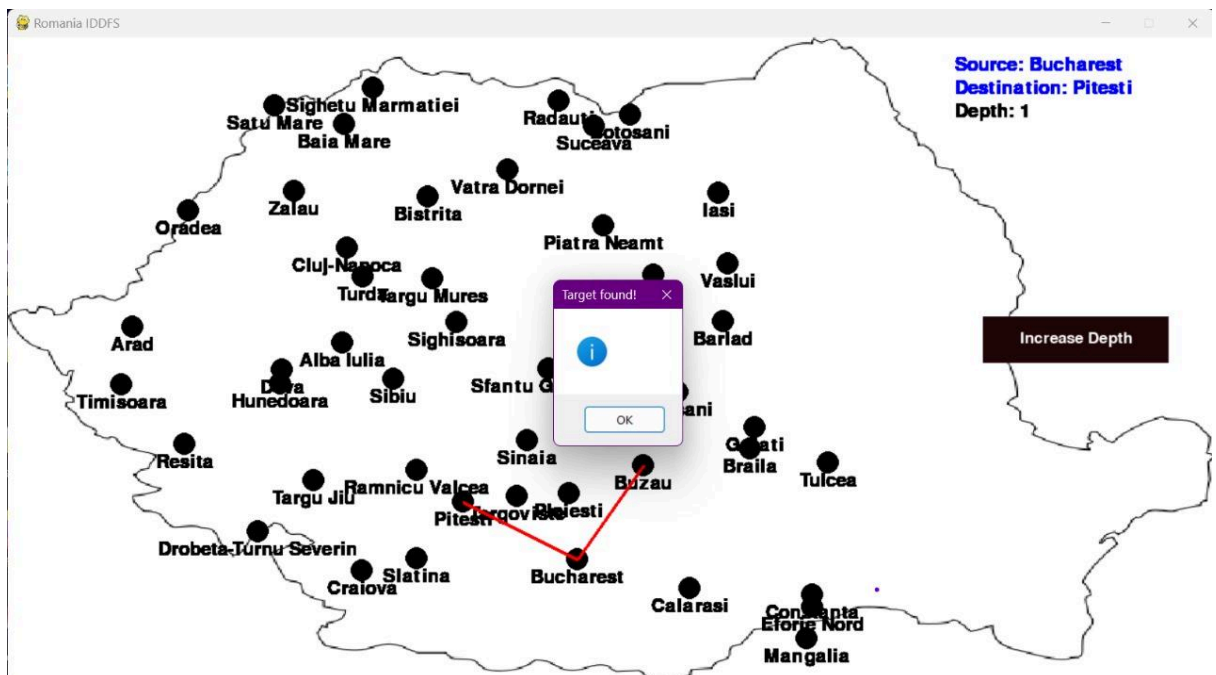
Type of Traversal: idfs

Enter Source: Bucharest

Enter Destination: Pitesti

Submit

```
PS C:\Users\Aman Joharapurkar\OneDrive\Desktop\AI Project\IDDFS-Simulation-master> python main.py
pygame 2.5.2 (SDL 2.28.3, Python 3.11.0)
Hello from the pygame community. https://www.pygame.org/contribute.html
Source Coordinates: 613 561
Destination Coordinates: 490 499
Bucharest -- Buzau
Bucharest -- Pitesti
```



**Components:**

- The program consists of a GUI interface created using Tkinter for user input.
- It utilises Pygame for visualising the map and traversal process.
- The IDDFS algorithm is implemented to find the path from the source to the destination.

**Limitations:**

- Support for other traversal algorithms could be added to make the program more versatile.
- The map data could be loaded from an external file to allow for easier customization and scalability.

**Conclusion:**

The implementation of the Iterative Deepening Depth-First Search (IDDFS) algorithm in this code provides a clear illustration of its effectiveness in traversing a graph, demonstrated through a map of Romania. By combining Pygame for visualisation and Tkinter for user interaction, the program offers an interactive learning experience for understanding graph traversal algorithms.

While the code effectively showcases the IDDFS algorithm's capability to find a path from a source city to a destination city, there are areas for improvement. Enhancements such as supporting additional traversal algorithms, loading map data dynamically, and improving the GUI for better user experience could elevate the program's utility and educational value.

In conclusion, this code serves as a valuable starting point for exploring graph traversal concepts, and with further refinements, it could become a powerful educational tool for students and enthusiasts interested in graph algorithms and problem-solving techniques.