SHORTEST PATH ALGORITHMS IN TRANSPORTATION MODELS : USING SEARCH ALGORITHMS ON ROAD NETWORK CONNECTION

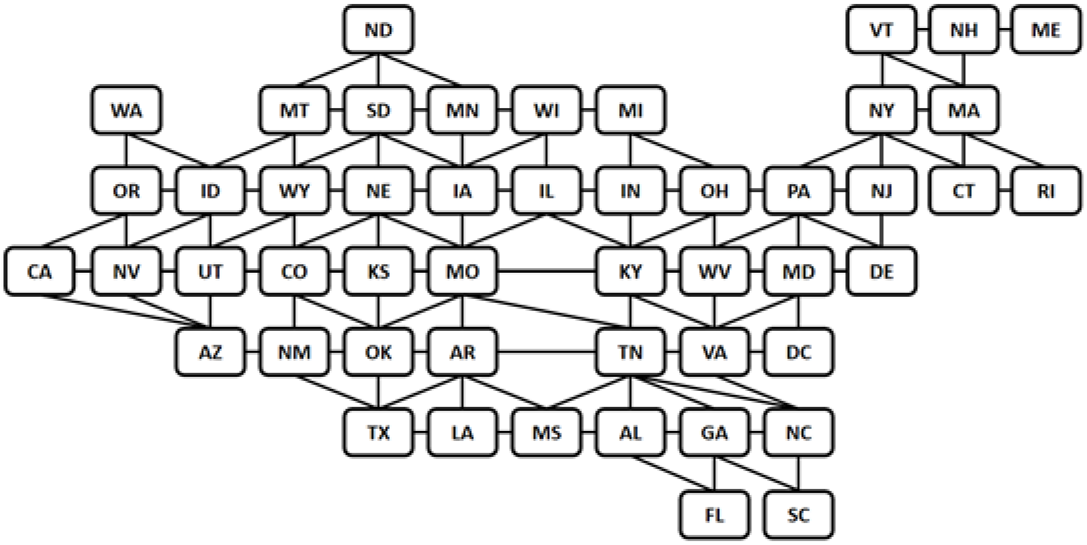
GROUP MEMBERS:

YASWANTH YADAV KORRAYI	-	18CE3AI18
RAKSHIT RANA	-	18AG32003
SANGEETH NELAPATI	-	18EC35046

PROBLEM STATEMENT:

Problem description:

Consider the graph presented below. Each node represents a single state of the US. If two states are neighbors, there is an edge between them.



Assuming that edge weights represent driving distances between state capitals. Here the problem is to find the shortest path between any two cities.

Importance and background:

Shortest Path Problems are among the most studied network flow optimization problems, with interesting applications in various fields. One such field is transportation, where various kinds of shortest path problems need to be solved. Since no “best" algorithm currently exists for every kind of transportation problem, which are able to capture the peculiarities of the problems under consideration. The aim of this work is to present in a unifying framework both the main algorithmic approaches for solving the shortest path problems that arise most frequently in the transportation field, and some important implementation techniques which allow efficient procedures.

Problem Statement:

input/output specification:

`driving.csv` - with **driving distances** between state capitals.

`straightline.csv` - with **straight line distances** between state capitals.

Numerical data in both files is either:

- a non-negative integer corresponding to the distance between two state capitals
- negative integer -1 indicating that there is no direct “road” (no edge on the graph below) between two state capitals.

This program accept two command line arguments corresponding to two states(initial and goal states) so code could be executed with

```
python main.py INITIAL GOAL
```

where:

`main.py` is python code file name,

`INITIAL` is the name of the initial state,

`GOAL` is the name of the final state.

Example: `python main.py WA TX`

AI Modelling(searching problem):

Here we mapped the problem statement to searching AI problems.

Task is to implement two informed search algorithms in python:

- Greedy Best First Search algorithm
- A* algorithm

and apply them to find a path between two state capitals using driving.csv and straightline.csv files.

Solution Approach:

Run Greedy Best First Search and A* algorithms searches to find a path between `INITIAL` and `GOAL` states and measure execution time (in seconds) for both methods.