

# Artificial and Computational Intelligence

## Assignment 1

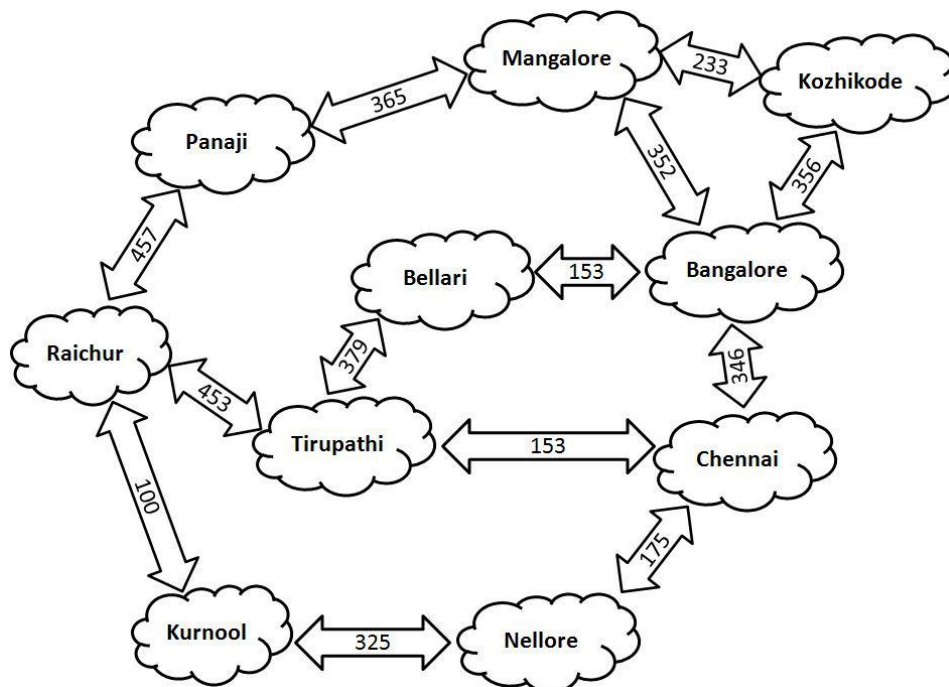
### Search Agent

#### Problem Statement: 19

Consider Vani, a business analyst working remotely from her hometown in City A due to the pandemic. She needs to urgently return to City B as her presence is required for an important project meeting. However, all flights from City A are fully booked due to a local festival, leaving Vani with no choice but to use road transport. Unfortunately, a sudden technical issue with Google Maps has rendered it unavailable. Help Vani plan the most optimal route using your knowledge of informed search algorithms.

Use the following algorithms to solve the problem:

1. A\*
2. Uniform cost Search



Answer the following:

1. Explain the PEAS and Task environment of the agent [20% weightage]
2. You decide to use the 'haversine' formula to calculate the great-circle distance between two points – that is, the shortest distance over the earth's surface between two points. Using the below latitude and longitude data for the cities, create a function which calculates the heuristic distance from each city to the destination city (refer link <https://www.movable-type.co.uk/scripts/latlong.html> for more information on Haversine formula). [30% weightage]

City	Latitude	Longitude
Panji	15.4909° N	73.8278° E
Raichur	16.2076° N	77.3463° E
Mangalore	12.9141° N	74.8560° E
Bellari	15.1394° N	76.9214° E
Tirupati	13.6288° N	79.4192° E
Kurnool	15.8281° N	78.0373° E
Kozhikode	11.2588° N	75.7804° E
Bangalore	12.9716° N	77.5946° E
Nellore	14.4426° N	79.9865° E
Chennai	13.0827° N	80.2707° E

3. Read the City A (source) and City B from the user. Use appropriate data structures and implement search algorithms (informed and local search) to find the path that covers all the **city/node/** with shortest **distance/cost** in the city as provided in the graph. Your output should contain the path taken by the agent (e.g. Panji, Raichur, Kurnool, Nellore, Chennai) and the total cost incurred in taking that path. [30% weightage]
4. Find and print space and time complexity using code in your implementation. [20% weightage]

NOTE:

- You are provided with the python notebook template which stipulates the structure of code and documentation. Use well intended python code.
- Use separate MS word document for explaining the theory part [PEAS]. Do not include theory part in the Python notebook except Python comments.
- The implementation code must be completely original and executable.
- Please keep your work (code, documentation) confidential. If your code is found to be plagiarized, you will be penalized severely. Parties involved in the copy will be considered equal partners and will be penalized severely.