**CSCE 3601: Artificial Intelligence**

**Assignment #1: Searching for Paths**

Due: Wednesday, March 4, 2023

**1. Introduction**

The problem you are given is to write a program to find a path between two European cities, using only the roads provided on the map. For instance, we may want you to find a path from Paris to Vienna. One such path would be Paris to Genoa to Rome to Munich to Prague to Vienna, with a total length of 629 + 328 + 582 + 174 + 185 = 1898 kilometers.

**2. The data**

You will need to propose a suitable data type to store the information in the map together with methods to generate the neighbor cities, test for the Goal State, and any other methods related to the domain. You will also need to implement a queue data type to store the nodes during the search. SEARCH also keeps track of how many nodes have been expanded altogether, and returns that information at the end, along with the path it found and the cost of the path. The final result of a search from Amsterdam to Berlin might be

**(14 520 Amsterdam Hamburg Berlin)**

indicating that a total of 14 search nodes were examined, that the total cost of the solution found was 520, and that the solution discovered is Amsterdam to Hamburg to Berlin.

The roads connecting the cities are provided in the accompanying text file: **roads.txt**, where you’ll find a tab-separated list of triples, each is a list in the form “city1 city2 dist” where "city1" and "city2" are connected by a direct road of length "dist" kilometers.

For example: **Copenhagen Hamburg 180** means the distance between Copenhagen and Hamburg is 180 km

Also provided are the cities’ coordinates in another accompanying text file: **positions.txt**, where each element is a tab-separated three-element list. The first element is the name of a city and the second and third elements are the “x-coord, y-coord” of the city, accordingly.

For example: **Copenhagen 687.7241 1323.35** means the position of Copenhagen is (687.7241, 1323.35) km

**3. The search functions**

It is required to experiment with:

1. Depth First Search,
2. Breadth First Search,
3. Iterative Deepening,
4. Uniform Cost search,
5. Greedy algorithm,
6. and A\* Search algorithm

You can use the air distance (aka Euclidean distance) as the heuristic function for estimating the distance from a certain node to the goal.

**4. User Interface**

You also have to implement a very simple user interface that will interact with the user asking for:

1. the city they want to start at
2. the city they want a path to
3. and the search strategy that they want to experiment

and return the resulting solution together with statistics. The statistics include total number of nodes visited, and cost; as mentioned previously:

**(14 520 Amsterdam Hamburg Berlin)**

**5. Your assignment (what you must hand in)**

**(a)** Run the search program on the test case "Paris to Vienna" using the six search methods, and compare their results and performance. An algorithm may be deemed successful according to several different measures. You are required to use two measures to compare the algorithms which are the number of nodes it expands, and the cost of the solution path it finds.

**(b)** Try to find city pairs for which breadth-first search expands fewer nodes than depth-first, and vice versa. Comment on your findings

**(c)** Try to experiment by decreasing and increasing the heuristic function by multiplying it by these factors 0.5,1.0 and 1.5 and notice the impact of the heuristic function on the nodes visited and cost.

**(d)** Submit a report on BB that contains: the documented source code of the functions you have written , the results of running your program as described above, and your written commentary on each run

**6. Grading**

The assignment will be evaluated as follows:

40% on the correct functionality of the program

20% on style and documentation of the written functions

40% on comments on the results

1. **Auxiliary Files**

europe\_map.png

positions.txt

roads.txt