Uniform Cost Search (UCS)

The UCS algorithm is a tree/graph search algorithm used for traversing or searching a weighted tree/graph. It expands the least cost node first, and is optimal when all costs are positive or zero. The algorithm uses a priority queue as its main data structure, where the node with the least path cost is given the highest priority.

In this project, UCS was used to find the shortest path (i.e., the path with the least total distance) between two cities.

Implementation

The Python application developed for this project consists of three main functions:

build_graph(path): This function reads the city data from a csv file specified by the path parameter and builds a graph as a dictionary. Each key in the dictionary is a city, and the corresponding value is a list of tuples representing neighboring cities and distances to them.

uniform_cost_search(graph, start, end): This function performs the UCS algorithm on the graph, starting from the start city and aiming to reach the end city. The function returns the total distance of the shortest path and a list of cities representing the path.

main(): This function prompts the user to enter the path of the road map file, the start city, and the target city. It then calls the other two functions to find the shortest path and its total distance, and prints the result.

The application is robust and can handle exceptions such as *FileNotFoundError* and a custom *CityNotFoundError*.

Testing

1) İstanbul – Kayseri
    a. Enter the path of the road map file: C:\Users\PC\Documents\Python\CSE358 Projects\Project-1\data\cities.csv (I will not show this line again in sake of simplicity.)
    b. Enter the start city: İstanbul
    c. Enter the target city: Kayseri
    d. The shortest path from İstanbul to Kayseri is: İstanbul -> Eskişehir -> Konya -> Kayseri
    e. The distance is: 435 km

2) Trabzon – İzmir
    a. Enter the start city: Trabzon
    b. Enter the target city: İzmir
    c. The shortest path from Trabzon to İzmir is: Trabzon -> Samsun -> Ankara -> Eskişehir -> İzmir
    d. The distance is: 525 km
3) Çanakkale – Konya
    a. Enter the start city: Çanakkale
    b. Enter the target city: Konya
    c. The shortest path from Çanakkale to Konya is: Çanakkale -> İstanbul -> Eskişehir -> Konya
    d. The distance is: 375 km
4) Balıkesir – Adana
    a. Enter the start city: Balıkesir
    b. Enter the target city: Adana
    c. The shortest path from Balıkesir to Adana is: Balıkesir -> İzmir -> Muğla -> Antalya -> Adana
    d. The distance is: 490 km
5) İstanbul – Paris
    a. Enter the start city: İstanbul
    b. Enter the target city: Paris
    c. Paris does not exist
    d. Paris

Application correctly handled the Turkish characters in the city names and provided the correct path and total distance. This was by far the biggest challange for this assignment. I spent 20 minutes until I realised I need to encode csv file as utf-8 and another 30 only to realize I need to include encoding in the Python code itself.

Application also correctly handled the case when the specified start or target city was not found in the road map, by raising a CityNotFoundError.