CS 225 - Final Project Goals

Tara D'Souza, Anirudh Avadhani, and Jacob Rubin

Dataset Summary

We will be using the OpenFlights dataset (https://openflights.org/data.html). The Airport dataset stores data about airports from around the world. The data is stored as separated strings, for instance: *507,"London Heathrow Airport","London","United Kingdom", "LHR", "EGLL", 51.4706,-0.461941,83,0, "E","Europe/London","airport","OurAirports".*

The relevant data in this string is the airport name, latitude, longitude, and city. In this string, that is London Heathrow Airport, 51.4706,-0.461941, and London. We will also use the Routes dataset to create a graph of the flight routes, in order to find the most efficient route between two airports. The data is also stored as separated strings, for example: *BA,1355,SIN,3316,LHR,507,,0,744 777*. The important data from this string is the source airport, source airport ID, destination airport, and destination airport ID. In this string, that is SIN, 3316, LHR, 507.

Graph Implementation

To implement this dataset, we will use a graph. In this graph, each node will represent an airport and store information such as the name, latitude, longitude, and city. Edges between nodes will represent a viable flight route across two airports, weighted by the distance between the airports.

Traversal

To traverse the graph, we will use a Binary First Search algorithm (BFS). This algorithm is more applicable to the flight dataset as we will traverse nodes, or airports, close to each other first. We will do this by visiting first all the adjacent nodes of a particular node. This is opposed to a Depth First Search algorithm where we would traverse to the farthest node and then backtrack.

Algorithms

We aim to implement two algorithms that will (1) find the shortest path between two airports as well as (2) find the shortest travel between two airports while travelling through a third airport.

1. **Dijkstra's Algorithm - Shortest Path Algorithm**
   We will use this algorithm to find the shortest path between two nodes. This algorithm is preferable to the Floyd-Warshall Algorithm because with Dijkstra's Algorithm we can find the shortest path between specified nodes rather than all nodes in the graph.

2. **Landmark Path Algorithm**
   To find the shortest landmark path, we intend to use an extension of the shortest path algorithm. To find the shortest path between A and B that visits C, we can find the shortest path from A to C and then from C to B.