Isaac Tong
Irfaan Attarwala
Eric Jin
Pranshu Chaturvedi

CS225 Final Project Goals

Project Goal:
The objective of the final project is to calculate the shortest distance between two airports. The user would input their starting airport, ending airport, and the maximum number of connections they are willing to take. Our program will then create a graph that starts from the starting airport and determine if the ending airport can be reached with the maximum number of connections specified by the user. If it exists, our program will proceed to use Dijkstra's shortest path algorithm to see which path requires the shortest distance as there may be multiple paths between the two airports. A graphical representation of the graph will then be outputted as an image showing all the possible routes to the destination and will ultimately highlight the route with the shortest distance.

Data Set Chosen:
We have chosen the OpenFlights dataset from the given dataset options. We plan to use the data on the possible flight connections between each airport as well as the distance between airports—which will act as weights for the edges of the graph. Since the OpenFlights dataset contains all possible routes served by all airlines, a graph can be constructed relating airports based on whether a route connects the two airports together. Furthermore, we will utilize the OpenFlights airport data to calculate the distance between the airports. Since the dataset provides the latitude and longitude of each airport, we can easily calculate the distance between airports. In summary, the nodes of the graph are airports, which can be terminal or connecting destinations for the user based on the imputed target location.

Algorithms/Pathfinding Methods Chosen:
Breadth-First Search:
As our program lets users choose two airports and the maximum number of connecting flights, we intend to conduct a breadth first search to check if it is possible to travel between the two airports given the maximum number of connecting flights. The breadth first search will be conducted on a graph we create that starts at the starting airport and whose maximum distance from the starting airport is the maximum number of connecting flights.

Dijkstra's Algorithm:
Once we use BFS to confirm that a path between the two airports exists, we plan to use Dijkstra's to determine what the shortest path between the two airports is. Shortest is defined as the minimum distance between the starting and destination airports, or the minimum sum of the weights of the edges between the starting and destination airports.

Graphical Representation:
We will overlay the nodes and edges of the graph onto a world map. This would represent every possible connection between airports. Next, depending on the user input, the graph will also highlight the optimal path crossing through all the relevant nodes from the starting and target airports in a different color. We plan to use a C++ GUI library to achieve this graphical output.