Project Goals:

Our goal is to write a program that allows us to find the shortest path between two airports that avoids hub airports, as defined by the betweenness centrality theory. The starting and ending airports can be hubs, but none of the intermediate airports can be. This would be useful for eccentric people who want to take the scenic route or just explore new airports they have not been to and love wasting time and money.

https://openflights.org/data.html

- Our chosen data set is from OpenFlight. We will mostly use the airports and routing data. The data comes in a .dat format.
- We will use a **betweenness centrality** measurement to find the hubs by finding which airport is most centrally located.
- When the user inputs an airport, we will use **BFS** to locate the starting and ending node on the graph data structure.
- When the two airports are located, we will remove the most central (hub) airports from the graph data structure and use **Dijkstra's algorithm** to find the most efficient routing. We will assume that the user does not care about layovers, and only wants to avoid hub airports.
- Our output to the user would be the airports in order that they would take, or if the route was not possible.
- A possible extension to the project would be to compare different betweenness centrality and path finding algorithms to each other to find the most optimal one.