Team Name: Project 3 Group 43

Team Members: Nikhil Sangamkar, Ryan DeMario, Kian Mesforush

Project Title: Skylink

Problem: We are trying to help users plan the most optimal route to take when planning their airline flights. There are often many ways to get to the same destination using different combinations of connecting flights. This project will provide the user with the best and most reliable flight options based on historical flight delay data.

Motivation: This program can help users reduce their chance of running into complications and delays – or even missing flights due to delayed layovers. I know from firsthand experience that certain airports tend to have more delays than others. If I knew more information about each airport before planning my flights, I can make better informed decisions about which path to take.

Features: The problem will be solved when the user is able to input a departure and a destination airport and receive an ordered list of the best airport routes to take to reach the destination airport. This list will be ordered on the fastest flight route, with a consideration for average flight delays that these airports have. The program will use two different graph algorithms to find the shortest path in two different ways and will show the results of using each algorithm for comparison.

Data: We will be using the Corgis public data set on airline flights and their delays (https://corgis-edu.github.io/corgis/json/airlines/). From this we can get information about the number of delays, reasons for delays, duration of delays, flights cancelled, and more. From there, there are many methods to calculate a delay weight. We could use a simple average for each airport's overall delays, or take a more complex route factoring delays for specific travel routes and creating a custom algorithm that considers different types of delays (i.e. national, weather, maintenance, etc)

Tools: We will be using C++ to build the backend of the program as this is a language that we are all familiar with. For the user interface, we will also use C++ and use the SFML library to create a frontend. We're using the SFML library because this is something we are all familiar with from previous classes.

Strategy: Data structures we're planning to use:

- Adjacency List: Each airport maps to a list of outgoing edges (flights)
- A*
- Djikstra
- Comparing A* vs Dijkstra's algorithm

Distribution of Responsibility and Roles: For this project the group discussed that we would be split up into three different roles, however, some of which intertwine with others so that if one member needs help the group as a whole can help that individual to complete the project. The roles we came up with are as follows,

- Front end developer, this person will deal with the interface and how the user interacts with the website or interface
- Back end developer, this person will code a lot of the backend which is what the interface will run off of creating the functionality of the program

- QA Engineer and Peer Editor, this role is built to help the two above and make the overall code more efficient while also helping either of the developers but is also tasked to make sure the code runs efficiently and timely, while creating test cases.

References:

- https://corgis-edu.github.io/corgis/json/airlines/
- https://www.sfml-dev.org/
- https://www.amexiogroup.com/2023/08/25/7-essential-roles-in-a-web-development-team/
- https://www.graphable.ai/blog/pathfinding-algorithms/
- https://www.geeksforgeeks.org/dsa/adjacency-matrix/