This project, first of all, really stressed me out and made me apply different concepts in Computer Science and not only what has been taught in this class. Initially, I had a difficult time trying to understand what was expected of me. I used the computer science way of solving problems, which was to break down the problem into the most basic level possible. I looked through the data files and mapped out the particular fields in the different CSV files that would contribute to the success of my implementation.

Before I could even start solving the problem, I had to read the files. That wasn't so much of a challenge because we had treated it in class, and how to do that was explicitly in the textbook. Thereafter, I had to think of a data structure to organize my data. Because the data had many records, over 7000, I knew looping through to get the specific information needed was not efficient in terms of time complexity. Instead, I used a hashmap because, with a provided key, I will retrieve the corresponding values in constant time.

After storing the data in a hashMap, it was now left with the search algorithm. This was the toughest part of the assignment. I initially tried implementing a graph, which took almost forever because I simply could not wrap my head around it since it got too complex.

After hours of brainstorming, I finally decided to borrow a concept discussed in my Intro to AI course. We implemented the Breadth First Search and used it to solve problems such as 8 puzzle. I started thinking of how to implement the algorithm concerning the problem I had at hand. The BFS takes in two parameters, the start location and the goal (end location). I created a node inside the BFS made up of the parent, successor state (airport code), stops, and airline code. There was a frontier that stored all the successor states (the next available flights), and an explored set that tracked all the visited states. The way the BFS works is that, while the frontier is not empty, it pops the first element and explores it. After the destination has been reached, it returns a solution path of all the nodes it visited to get there.