University of San Diego

Project 3: Flight Path Finding Algorithm

Om Kanwar

Data Structures and Algorithms Spring 2018

Dr. Jiang

May 3, 2018

**Assertion Statement**

I, Om Kanwar, assert that I have submitted a complete and functional project. Our program runs without failure or error and is able to handle any form of flights.txt file and cities.txt file as per the assignment specifications. The program first reads in the cities and flight paths from the aforementioned text files in the directory and then prompts the user to input an origin city and a destination city. After the user inputs these cities, the program finds a path from the origin city to the destination city, if one exists, and calculates the cost of the flight plan per the input text files and outputs a summary of the flight path to the terminal. I did not submit the code before midnight on May 1st, 2018, but early the next day on May 2nd, 2018. I unfortunately had some bugs in my algorithm and continued to get Null Pointer Exceptions and Index Out of Bounds exceptions that I could not figure out before the deadline passed. After I created a completely functional program with the correct algorithm implementation, I submitted my code. This project folder was submitted at the beginning of class on May 3rd, 2018.

**Project Summary**

*Description of Java Classes:*

* **City.java** – This class initializes the constructor for the City class and helps implement the cityPath class. This file contains a String variable to hold the name of the cities and a Boolean variable to check if a specific City has been visited.
* **cityPath.java** – This file navigates the paths of the origin city entered to the adjacent cities. This class uses the City class and creates City variables for an origin city and destination city and continuously updates the destination city as the program runs through a path from an origin to a destination.
* **Driver.java** – This class is the main driver for the program. The file implements the files cityPath and City and uses the functions and the variables within those files to find a path from a user inputted source city to a user inputted destination city. The main method in this class initializes array lists and a hash map to take in data from the files and hold them for use later in this program. The hash map holds a key field with the cost of each flight. The main use of the hash map in this program is to help calculate the total cost of the flight path from the origin to the destination. The getStartingCity() and getDestCity() functions simply prompt the user to input a source and destination city respectively and checks to make sure the inputted city is within the given text files. The searchCity() function updates the current city to be checked when navigating through a flight path to a destination city. The getCities() and getFlights() functions are simple functions to read in the data from the given text files and make sure that the correct files are within the scope of the directory. The main algorithm implementation is also contained within the main method in a try/catch block of code. The algorithm is implemented in a while statement to loop continuously until a path has been found from a source to a destination. This section of the code also catches exceptions if there are no flights available from a given origin city or if there is no path at all from a given origin city to a destination city. Finally, the last portion of the main method prints out to the user the overall flight plan to the user after finding the path and calculating the cost of the path.