Project 5

Alexander J Sanna

Due: December 8th, 2022

Project Specification

For this project, I implemented the following classes:
DictionaryInterface<T>
VertexInterface<T>
BasicGraphInterface<T>
StackInterface<T>
AirportApp
AirportInformation
Airport
Digraph

The javadoc files for the interfaces are attached below, as for the others:
AirportApp was the main method. It ran the entire program and served as the client. Inside it **held a dictionary where the key was the airport code and the value was the AirportInformation object.** The object held a reference to the name, city and state that the airport was in. This was an incredibly efficient way to store the data because it ensure for o(1) searches for data.

The Airport class was the implementation of the VertexInterface and the Digraph was the implementation of the basic graph interface. The digraph had a dictionary that also held the code as the key but this time the vertex interface was the value correlated with the code. Again, an extremely efficient way of coding this to ensure that each search was O(1).

The Airport class also held an arraylist reference to the edge object. The edge was an inner class in airport that served as the connections between vertices.

The DiGraph class also had an inner class called EntryPQ which implemented the comparable interface. This was so that in the priority Queue we could compare these objects as they held references to weight of edges between 2 referenced vertices.

GET SHORTEST PATH was also implemented and used the entrypq in a priority queue to ensure the return of the shortest path. Also gave path in a StackInterface. This was incredibly useful in transferring data.

The DIGraph also has a reset method to ensure every visit flag is reset everytime the get shortest path method is run. No errors.

Testing Methods

The testing of these methods consisted of the main method being run multiple times with different parameters.

It can successfully give the distance between 2 locations
Return information about a given location
And insert a connections between two locations with a distance x.

The only issue i ran into was that the flags werent resetting with each use of the D command, so i implemented a private reset method to fix this.

What I learned

What I learned? I learned more about everything with this project, it was a lot. I had to use basically everything I had learned in this class to get it close to working but I am super proud of it. It uses two dictionaries to store data, stacks to transfer data between methods and graphs to store vertices and their respective edges. This was a huge task but came out great in the end.