CSE2011 Summer 2016 - Assignment 3

Due date August 9th at 2:00PM. No late submissions will be accepted.

You must work individually. The assignment is worth 5% for part 1 and 5% for part 2. The code must be properly structured and documented. Your code will be tested by an automatic program.

Submit the source code for all programs and answers to any questions given in the assignment. The submitted source code is expected to use good coding style and be properly commented. This includes proper javadoc comments (including @param, @returns, etc.). We should be able to produce suitable api's using the javadoc command. Do not place any .java files in a package, i.e. do not use a package statement at the start of your files. Zip all of your .java files and any other files you wish to submit (do not submit any .class files) into a file called a3.zip and submit it using the command

```
% submit 2011 a3 a3.zip
```

Part 1

Create generic data structure(s) for storing graphs using the adjacency list structure. The vertices will store objects of generic type V and the edges store objects of generic type E. The entries should be location-aware. The data structure must support adding and removing of given vertex or edge. For quick reference the vertices should also be stored in a hash table keyed by vertex's object V. Follow the diagram from the lecture slides.

Part 2

Create a program that will find the shortest connection travel time between airports. It will accept the following commands:

```
+ YYZ JFK 120 plane
```

to add a connection from airport YYZ to airport JFK that takes 120 minutes using a plane

```
- YYZ
```

to remove an airport from the database

```
- YYZ JFK 120 plane
```

to remove a connection from airport YYZ to airport JFK that takes 120 minutes using plane

```
? YYZ
```

to list all connections from YYZ (lines in the format YYZ JFK 120 plane)

```
? YYZ LAX
```

to find the quickest route from YYZ to LAX, it shout print the total duration and then list the individual connections for this option (lines in the format YYZ JFK 120 plane)

```
?
```

to list all connections in memory (lines in the format YYZ JFK 120 plane)

QUIT

to end the program

You must use the data structure developed in part 1.

Marking Scheme

50% = working program / all requested features in code

10% = code robustness, provisions of unexpected data/input

10% = proper comments

10% = proper coding style, proper structure, use of interfaces etc.

10% = show that you tested the code using typical and boundary (unusual) data

10% = marker’s discretion