CS 245 - Assignment 3 Summer, 2021

Assignment 3 - Road Trip

The purpose of this assignment is to demonstrate your mastery of Graph algorithms by extending an existing algorithm for a unique purpose.

Background

Design and implement an application for scheduling road trips in the continental USA — 48 of the states, excluding Alaska and Hawaii. Assume the user will supply a starting city (San Francisco, for example) and an ending city (eg. Miami). In addition, the user will supply zero, one or more events or places of interest by name, such as the Grand Canyon, Paul Bunyon and Babe the Blue Ox in Minnesota or the Statue of Liberty in New York. Once implemented, your solution must find a route from the starting city to the ending city in a way that visits each of the events or places of interest while keeping the number of miles driven to a minimum. In the example from above, the solution must return a route from San Francisco through the following places, in order:

- Grand Canyon AZ
- Bemidji MN
- New York NY

..., in that order, finally ending in Miami.

With this assignment, you are supplied two files in CSV format, as described in the Table below.

File	Description
attractions.csv	A file with names of events or places of interest. The user may choose to stop at zero, one or more of the places listed in this file. The file is in CSV format with two columns, in order: ● Event or place of interest ● Location (city state)
roads.csv	A file with names of cities and the distances between them. The user must choose one starting city and one ending city. The file is in CSV format with four columns, in order: ● Location (city state) ● Location (city state) ● Distance (miles) ● Distance (minutes)

The roads.csv file contains several road distances between nearby cities and towns of a sufficiently large size for many parts of the USA. For example, this file contains entries as follows:

- Point Reyes Station CA,San Francisco CA,42,67
- Sacramento CA,San Francisco CA,88,81
- Salinas CA,San Jose CA,59,56
- San Francisco CA,San Jose CA,48,52
- San Francisco CA,Stockton CA,88,79

... among many others. One of the entries above indicates that the distance from San Francisco to San Jose is 48 miles and that it takes 52 minutes to travel between them. (This file will NOT contain distances which skip cities or towns. For example, it will not contain an entry for San Francisco to Salinas because of the city, San Jose), which lies in between.)

Part 1 - Design

Given the inputs:

- Starting city
- Ending city
- A List of events or places of interest

… show the UML for your proposed solution. The solution must have one function to determine the route with the signature similar to the following:

```
List<?> route(String starting_city,
     String ending_city,
     List<String> attractions)
```

This function could take the user-supplied starting city, ending city and list of attractions. (The list of attractions does not have a particular order.) This function returns a list representing the route the user should take. (Your design must include a description of the type or class contained by the returned list, hence the "?" character.)

In your description, you may be motivated by the following questions:

- What data structure(s) will you use to represent the data in the file “attractions.csv”?
- What data structure(s) will you use to represent the data in the file “roads.csv”?
- What algorithm(s) will you use to find the shortest route from starting city to ending city through all the specified events?

Your design should focus on speed. In other words, once the user has supplied a starting city, an ending city and a set of events or places of interest, your solution should return the shortest route in the shortest possible algorithmic running time.

Consider answering the following questions in your explanation:

- What classes would you use? What public or private functions would they have? (Feel free to use UML in your explanation.) How would these classes collaborate in your solution? Which class / function contains the "route" function?
- What data structures would you use? What algorithms on those data structures would be important?

- How will you store the data from the distance file? How will you store the data from the interests file?
- What is the running time of the "route" function? (How do you know?)

Your explanation only needs to describe your design.

You may seek (but do not need to have) approval of your design before proceeding to Part 2 - Implementation.

Part 2 - Implementation

Implement your design in one or more Java classes, including one main function. This main function will be provided with the locations of the two files (roads.csv and attractions.csv) as described above. This data must be read and stored by your implementation. Once read and stored, your implementation must accept one or more queries from the user. Where possible, the implementation must respond with a path from one city to another via all the attractions indicated by the user. An example of the interaction is as follows, with the implementation's outputs in bold and the user's inputs in non-bold:

```
Name of starting city (or EXIT to quit): San Francisco CA
Name of ending city: Miami FL
List an attraction along the way (or ENOUGH to stop listing): Grand Canyon
List an attraction along the way (or ENOUGH to stop listing): Nonsense
Attraction "Nonsense" unknown.
List an attraction along the way (or ENOUGH to stop listing): Boot Hill
List an attraction along the way (or ENOUGH to stop listing): ENOUGH
Here is the best route for your trip:
* San Francisco CA -> Stockton CA
* Stockton CA -> Fresno CA
```

… etc.

Submission

There are two submissions for this assignment:

1. Design — submit a document describing your plan. This design document can use any tools you wish (UML, narrative, etc.) but must be submitted as a Google Drive document or a PDF. If you have not received approval of this design document, your submission for part 2 (implementation) will not be accepted.
2. Implementation — submit one or more properly-document Java source files for the design described in part 1.

Do not submit the data files (roads.csv and attractions.csv) but you may also add any comments or notes in a README file (text or PDF document). Ensure you do the following:

- Check your code into your own github repository. If your github repository is not public, contact the instructor for users who must have access.
- On Canvas, submit the URL for your github repository.

Grading

Your grade for this assignment will be determined as follows:

- 30% = Design: The degree to which your designed solution represents a reasonable object-oriented or procedural decomposition to these problems using established or novel data structures and algorithms.
- 50% = Implementation: The degree to which your Java implementations runs successfully with the source files and data provided, producing the expected results.
- 10% = Efficiency: The degree to which your implementation is efficient with respect to algorithmic time and space. This includes, but is not limited to, any containers used to store data provided in this example or for a larger, similar dataset.
- 10% = Style: The degree to which your implementation is readable to the point of being self-documenting; in other words: consistency with respect to comments describing the purpose of each class and the purpose of each function within a class, descriptive naming of variables and functions, and commenting of any code which is not straightforward or is in any way difficult to understand.