
Semester Project
(CLO5->PLO3)

Submission date: 13th, June 2021

Submission time: 2359hrs

Note:

- Submission should be on LMS. Assignment submission through email will not be accepted. Submit your complete code in .docx file. Late Submission will not be marked.
- You are allowed to work in group of 3 (at max). Enter your group member's name

Do your own work. Any kind of plagiarism found will result in negative marks

Map & Route Planner

In this project you are asked to implement a software system that makes use of map data of Pakistan. In particular, you should strive to show fulfillment of the following course goals: Be able to choose among and make use of the most important algorithms and data structures in libraries, based on knowledge of their complexity.

1. Intra-city:

The system should allow a user to enter two addresses, each consisting of a street name and a number, and locate these in the data set (if the addresses exist). If several addresses match, a list of possible postcodes should be displayed, and the user should be prompted to select a postcode. Once the addresses have been uniquely determined, the system should compute and report a fastest path from the first address to the second one. As you will experience, however, it is impossible to determine exactly where an address is on a road segment. For that reason, you are allowed to start your search from one of the two road intersections that is adjacent to the start address (arbitrarily chosen) and end your search at one of the two road intersection adjacent to the goal address (again arbitrarily chosen). You may also make the following additional simplifying assumptions:



- There are no turn restrictions.
- The time to drive along a road segment is its DRIVETIME.
- Turning takes zero time.
- The format of the fastest path is a list of street names and distances (no turn information).

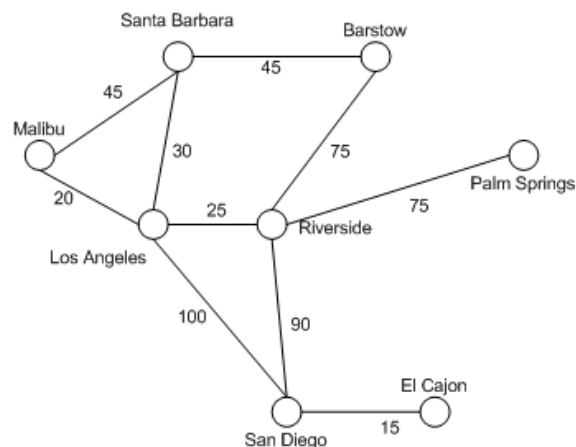
2. Inter-city:

Suppose user wants to know the route for Multan from Rawalpindi via Lahore, then the application should make a route which will show the major cities which will come along the way. You should also sort out shortest route possible.

3. Implementation:

A map can be easily modeled as a graph, with the cities as nodes and the roads connecting the cities as edges. If we wanted to determine the shortest distance and route from one city to another, we first need to assign a cost from traveling from one city to another. The logical solution would be to give each edge a *weight*, such as how many miles, cost and time consumption it is from one city to another.

Figure shows a graph that represents several cities in southern California. The cost of any particular path from one city to another is the sum of the costs of the edges along the path. The shortest path, then, would be the path with the least cost. In Figure, for example, a trip from San Diego to Santa Barbara is 210 miles if driving through Riverside, then to Barstow, and then back to Santa Barbara. The shortest trip, however, is to drive 100 miles to Los Angeles, and then another 30 up to Santa Barbara.



Note: Your application should be capable of finding a path either optimized with respect to time or distance as specified by user.

Perform Complexity Analysis of all functions of your project and compute Big-Oh. You have to follow complete procedure to compute $T(n)$.

Trait	Exceptional [10-8]	Acceptable [7-6]	Amateur [5-3]	Unsatisfactory [2-0]
Application Functionality 10%	Application compiles with no warnings. Robust operation of the application, with good recovery.	Application compiles with few or no warnings. Consideration given to unusual conditions with reasonable recovery	Application compiles and runs without crashing. Some attempt at detecting and correcting errors.	Application does not compile or compiles but crashes. Confusing. Little or no error detection or correction.
Specification & Data structure implementation 60%	The program works and meets all of the specifications. Well-designed and Excellent implementation of data structure. Providing all the required functions that might be useful and aids in performing the operations on data structure.	The program works and produces the correct results and displays them correctly. It also meets most of the other specifications. Good implementation of data structure.	The program produces correct results but does not display them correctly. Fair implementation of data structure.	The program is producing incorrect results. Worst implementation of data structure.
Reusability & Efficiency 5%	The code could be reused as a whole or each class could be reused.	Most of the code could be reused in other programs.	Some parts of the code require change before they could be reused in other programs.	The code is not organized for reusability.
Graphical User Interface and input validation 20%	A good Graphical User Interface Which enables the user to interact with the application easily. With all the input fields properly validated. A message displayed to user in case of wrong input entered by user and prompt till user enters the correct input.	A satisfactory Graphical User Interface . Good design. Error Message displayed to user in case of incorrect input entered by user. The program continues execution without prompting again till the correct input is entered	A Graphical User Interface With not an impressive interactive pane. Some of the inputs are validated leaving the others.	No user friendly Graphical User Interface or a Graphical User Interface with no input validation.
Documentation and Readability 5%	The documentation is well written and clearly explains what the code is accomplishing and how along with the results. Code is well commented. The code is exceptionally well organized and very easy to follow.	The documentation is well written but the output is missing or the code is not well commented. The code is fairly easy to read.	The documentation does not follow the format and not a well commented code. The code is readable only by someone who knows what it is supposed to be doing.	The documentation does not follow the format, no output results and does not help the reader understand the code. The code is poorly organized and very difficult to read.

PROJECT ASSESSMENT RUBRICS