Assignment 3

Due Date: 31 May2024

Weighting: 40%

Group or Individual: Individual

In assignments 1 and 2 we have assessed your knowledge about linear data structures, linear data structure-based algorithm, non-linear data structures, and non-linear data structure-based algorithms and your skills in applying the knowledge in the design and implementation of reusable Abstract Data Types (ADTs). This assessment is different from the previous two assessments. In this assessment, we assess your knowledge and skills in applications of graph algorithms in a real-world software development project.

Since this unit is about data structures and algorithms, rather than object-oriented analysis or object-oriented design, we have already completed the object-oriented analysis and design fir you and have developed a skeleton of the software application. Your job is to implement some functionalities of the software.

The skeleton is an incomplete Microsoft Visual Studio 2022 (Community Edition) Console Application project. The software design and the skeleton will be explained in Week 9 lecture. In the lecture we will also demonstrate to sow what you are expected to do in this assignment. The lecture will be recorded and will be automatically uploaded to Echo360 on the same day.

It is assumed that you have knowledge and skills in object-oriented programming in C#, which are covered in the prerequisite of this unit, CAB201.

1. Information and Software Requirements

Imagine you are working as a computer scientist for a city's urban planning department. Your task is to develop a software application to assess the design of a transportation network, including the connectivity of the transportation network, the shortest distance between two given intersections in the transportation network, and the shortest distance between all pairs of intersections in the transportation network.

The transportation network is modelled as a directed weighted graph, where intersections are represented by vertices, and links between two intersections are represented by edges, and the distances between a pair of linked intersections is represented by an edge weight.

Below is a list of functional requirements that we have identified during the software requirement analysis:

- **Data Input** - Read information from a text file about a transportation network design into the system. Below is an example showing the structure of the text file:

  A, B, 3

  A, C, 4

  B, A, 4

C, A, 5

B, C, 1

C, B, 1

In the transportation network design, there are three intersections, A, B and C, and six roads (links), <A, B>, <A, C>, <B, A>, <C, A>, <B, C> and <C, B>. The distance of <A, B> is 3, the distance of <B, A> is 4. the distance of <A, C> is 4 and the distance <C, A> is 5, the distance of <B, C> is 1 and the distance of <C, B> is also 1.

- **Connectivity** – Design and implement an algorithm to check if a transportation network is strongly connected. A transportation network is strongly connected, if there exists a path from any intersection to all the other intersections in the transportation network. This functionality is useful as we need to know if one could travel from one intersection to any other intersection in the transportation network plan.
- **Shortest Distance between Two Intersections** - Design and implement an algorithm to find the shortest distance between two given different intersections. This functionality is useful as we need to know the shortest distance from one intersection to another intersection.
- **Shortest Distance between All Pairs of Intersections** – Apply Floyd's algorithm to find the shortest distance between all pairs of intersections. This functionality is useful as it can be used to examine the shortest distance between any pair of intersections.

2. Assignment Requirements

- The programming language used in this assignment must be C#.
- Your complete console application can be compiled and run in Microsoft Visual Studio 2022 (Community Edition) on Windows 10/11.

3. Submissions

- Your submission should be a single zip file named by *your-student-number*.zip archiving your whole console application project. You do not need to submit any other documents. The submitted archive must be in standard .zip format. Uploads in other formats such as .7z, .rar, .gz, etc, will not be accepted.
- Your submission must be submitted via the Gradescope. Email submissions are not accepted.
- You can submit your assignment unlimited times before the deadline.