Assignment 3

Due Date: 2 June 2023

Weighting: 40%

Group or Individual: Individual

In assignment 1 we have assessed your knowledge about linear data structures and linear data structure-based algorithms and your skills in applying the knowledge in ADT implementations. In assessment 2, we have assessed your knowledge about non-linear data structures and algorithms and your skills in applying the knowledge in ADT implementations. This assessment is different from the previous two. In this assessment, we assess your knowledge about data structures and algorithms and your skills in applying the knowledge in developing a real-world software, rather than in ADT implementations.

In this assignment, you need to identify significant ADTs in a real-world application, design and implement the identified ADTs, and then use the ADTs to build a real-world application. There are some computational problems arising from the ADT implementations and application building. Thus, you also need to apply some algorithms that you have learnt in this unit to solve those computational problems.

Since this unit is about data structures and algorithms, rather than object-oriented analysis or object-oriented design, we will not assess your knowledge or skills about object-oriented analysis or object-oriented design. Instead, we assess your knowledge about data structures and algorithms and your skills in applying your knowledge about data structures and algorithms to software development.

In this assignment, we assume that you have knowledge and skills in object-oriented programming in C#, which is covered in CAB201, the prerequisite of this unit.

1. Information and Software Requirements

You are hired to develop a project management system to manage a list of tasks in a project. There may be some dependencies among these tasks in the project. For example, let's say you are managing a software development project. You have a list of tasks that need to be completed, such as design, coding, testing, and deployment. You know that some tasks are dependent on others, such as testing cannot start until coding is completed.

The information about the tasks and the dependencies among the tasks are stored in a text file. The information about a task includes a task ID, which is a string, and the time needed to complete the task, which is a positive integer, and a list of tasks that the task depends on. The information about one task is stored in a separated line in the text file. Here is an example, demonstrating the organisation of the text file:

T1, 100

T2, 30, T1

T3, 50, T2, T5

T4, 90, T1, T7

T5, 70, T2, T4

T6, 55, T5

T7, 50

The above indicates that the time needed to complete task T1 is 100 and T1 does not depend on any other jobs in the project, the time needed to complete task T2 is 30 and task T2 cannot commence until task T1 has been completed, the time needed to complete task T3 is 50 and task T3 cannot commence until both tasks T2 and T5 have been completed, and so on.

The project management system should have a Microsoft Console Application with a command line menu allowing the user to do the following:

- Ask the user to enter the name of a text file in which the information about the tasks in a project and the dependencies among the tasks are stored and read the information from the text file into the system.
- Add a new job with time needed to complete the job and other jobs that the job depends on into the project.
- Remove a job from the project.
- Change the time that a task needs to complete the job.
- Save the (updated) information about the tasks and dependencies back to the opened input text file.
- Find a sequence of the tasks that does not violate any job dependency and save the sequence to a text file, namely, Sequence.txt. For example, for the above sample project, the content in Sequence.txt should look like:

  T1, T2, T7, T4, T5, T3, T6
- Find the earliest possible commencement time for each of the tasks in the project. and save the solution into a text file, namely, EarliestTimes.txt. For example, for the above sample project, the content in EarliestTimes.txt should look like:

  T1, 0

  T2, 100

  T3, 260

  T4, 100

  T5, 190

  T6, 260

  T7, 0

3. Assignment Requirements

- The programming language used in this assignment must be C#.

- Your visual studio project type must be a console application of Microsoft Visual Studio 2022 (Community Edition).

4. 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 from CAB301 Canvas website. Email submissions are not accepted.
- You can submit your assignment unlimited times before the deadline.