Computer Science 3A

Mini Project

2023-03-08

Deadline: 2022-05-08 12h00 Marks: 100

You are required to implement a practical project that will demonstrate your proficiency with data structures discussed during this course (especially the Graph ADT) using the theme discussed below. The practical mini project must be submitted on or before the deadline (**No Exceptions**). *Late submissions will not be accepted and the student will receive zero for any late assignments.* This practical project will count as part of the practical component of this course (and 25% of your semester mark).

Note: — *A signed plagiarism assignment submission form must be submitted alongside the practical assignment that will cover this assignment. The appropriate form will be made available on EVE and there will be a separate submission area.*

The mini project demonstrates your understanding and proficiency with the concepts discussed in class and how they can be applied within a specific domain to solve a problem. You might need to make use of data structures we have not yet covered in class. In this case you will need to research the implementation of the appropriate data structure. The practical mini project must meet the following requirements:

1. **Programming Language:** Java
2. **Submitted File Format:** Zip (Only!)
3. **Submitted File Naming Convention:** *studentnumber*_miniproject.zip
4. The ZIP file must contain the following directory structure:
   (a) `src` – that contains all of the Java source files.
   (b) `dist` – that contains a executable `jar` file.
   (c) `ss` – that contains PowerPoint or PDF Slideshow for your Mini Project.
5. The use of third-party libraries for **primary** functionally is strictly **prohibited**, however, third party libraries can be used for other functionality such as communications and visualisation (such as normal JavaFX, JFreeChart, GraphStream, JGraphT, Yworks, JUNG or JMonkey).
6. Your assignment must be executable from a `jar` file. If the assignment cannot be executed, you will receive zero.

7. If the assignment is too big to upload, please upload the source files to Eve and contact the lecturers for the alternative upload method.
8. Your assignment must make use of a **graph-based structure** at its core.
9. The use of other data structures for auxiliary operations is encouraged (List, Stack, Queue, Heap, Dictionary, Trees, etc.)
10. You must write the data structures yourself, you may use the textbook to guide you in the implementation of your data structures.
11. You may not do a practical implementation that has already been assigned during the course.
12. Your project must be implemented as a desktop application based, with no data comms and APIs.
13. The assignment is an individual project and should not overlap with any other students (past or present) or source code found on the Internet. Each assignment will be checked and if found guilty will be sent up for disciplinary action.

Theme: Socio-Economic Problem Solving with Graphs

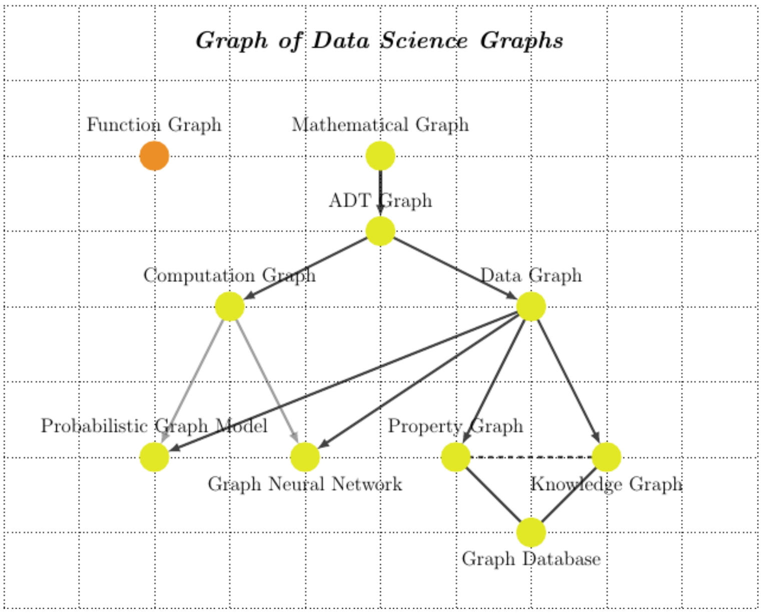
South Africa has a great deal of socio-economical problems that affect the everyday lives of our people. Technology can play a role to empower key stakeholders to potentially solve these problems, but much of these problems are especially complex to solve and require very nuanced solutions for them not only to be viable, but scalable too. One particular data structure that scales well are graph abstract data types (ADT). Therefore the theme for the Mini Project is solving South African-based socio-economical problems using graph-based solutions.

A (far too) simple definition of a mathematical graph is "a set of node or points together with lines joining certain pairs of these point"[1]. Informally put, it is the the enumeration of things (nodes) and their relationships (edges). More so, each of these edges can be directed or undirected, further constraining how relationships are encapsulated in a graph. More specifically in computer science we refer to the abstract data type graph, which is related to mathematical graphs, but is more focused on possible values and operations on these values, thereby making it more concrete and useful for problem solving. Overall, these graphs are useful when modelling the following:

1. Behaviour
2. Computation
3. Data

[1]Bondy, J. A., & Murty, U. S. R. (1976). Graph theory with applications (Vol. 290). London: Macmillan.

Key examples of how graphs are used in data science are shown in the figure below (taken from https://www.openriskmanagement.com/9_ways_graphs_show_up_in_data_science/):



This seemingly trivial idea can be used to describe and analyse many real-world situations, including resource allocation problems, non-repudiation in corruptable systems and optimising business problems. These problems are normally are abstracted in terms of a graph with the nodes representing resources, locations or people and the edges representing their relationships or cost. We can then use algorithms that operate on the graph to solve the problem. Examples of tasks, along with their associated algorithms that can be used on the graph include:

1. Shortest Path: as Dijkstra or Bellman-Ford
2. Cycle Detection: Floyd or Brent
3. Minimum Spanning Tree: Prim or Kruskal
4. Strongly Connected components: Kosaraju or Tarjan
5. Topological sorting: Kahn or DFS
6. Graph Colouring: Greedy colouring, DFS or BFS
7. Maximum Flow: Ford-Fulkerson, Edmonds-Karp or Dinic
8. Matching: Hopcroft-Karp, Hungarian or Blossom

The task for this mini project is to solve a South African societal problem using graphs to represent data in the system and algorithms that work on this graph. The idea is to solve a specific problem by creating a desktop application that serves as a proof of concept that this problem can be solved in some way usin a graph. Some of the key societal problems in South Africa and potential solutions include the following:

1. High unemployment
2. Poverty and Inequality
3. Limited access to public services

Your practical implementation **must** address a socio-economical problem and solve it using the **Graph ADT** provided as a primary component of an application. You are free to choose which problem you want to address, it just needs to use the Graph ADT provided.

Examples of mini projects you may **NOT** implement include (i.e. the ban list that will result in you getting zero):

1. **A Utility library** — where there is no user interface (remember we want to play with it and see it works).
2. **Anything copied from the Internet or a previous project**, e.g. High School students who struggle with making friends — This is plagiarism and the appropriate disciplinary action will follow should this occur (just don't do it).

You will receive marks based on the scope of your practical mini project, your use of the Graph-based social network, user interface and the presentation (you will get guidance on these throughout the semester).

You must confirm your individual project by **17th March 2022 at noon**. The method of confirming the project topic will be a Google Form (https://forms.gle/ioMsGiK848L2Y3Sc8) and the outcome (out of 5, where 2 and below is a rejection of your project) will be shown on Eve for each student's topic.

Marksheet

1. **Abstraction** (Successfully translates problem domain and aspects to social graph) [10]
2. **Use of a Graph-based social graph** (CRUD of nodes and edges, including profile, posts feed, reactions and notifications) [20]
3. **Logic and Complexity** (Can facilitate solution processes using graph algorithms, provides a dynamic graph and solves problem) [30]
4. **Novelty** (New or unusual choice of problem, attributes, etc.) [10]
5. **Look and Feel** (Aesthetics - A graphical user interface that facilitates the use of the social graph) [20]
6. **Video with Slideshow** (An 8 minute video describing their project with a SS that depicts all the necessary aspects and provides screenshots for processes) [10]