DEPARTMENT OF
SOFTWARE TECHNOLOGY

CCSDSALG (Data Structures and Algorithms)

Major Course Output 2: Graphs

Major Details

Groupings:	At most 4 members in a group
Deadline:	August 4, 2025 (M) 0800
Percentage:	20% of final grade
Submission Guidelines:	Submit deliverables in AnimoSpace
Filename Format:	`CCDSALG-MCO2-`*`Section-GroupNumber`*`.zip`

Deliverables

You are to submit the following files **separately** in AnimoSpace:

- A `source.zip` file containing a folder called `source`, which contains **all** the source codes for the project; and
- A PDF file called `report.pdf`, which contains the project report.

Project Specifications

SOCIAL GRAPH DATASET

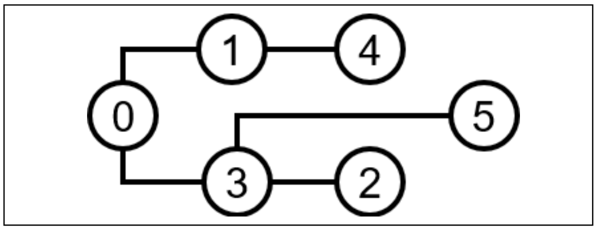
- For this project, you will work on a real-world dataset collected from Facebook (Traud, Mucha & Porter, 2005).
- The data is composed of complete Facebook networks of selected American colleges and universities. It was collected as a snapshot of a single day in September 2005. In this project, you can focus only on a few selected colleges or universities.
- The data describes the structure of the social network within a given college or university. It describes which accounts are friends with each other within the network. Names of the people are anonymized.
- The social network structure, otherwise known as the **social graph**, can be represented as a graph, where each node represents an account. An edge is present from node a to node b if and only if a and b are friends.
- For this project, the data has been converted from its original MATLAB (`.mat`) format to a text file for easier reading.
- You are provided several sets of data of varying sizes from selected colleges and universities. If you want more datasets, you can check out this link, but you have to convert them to `.txt` by yourself.

Traud, A. L., Mucha, P. J., & Porter, M. A. (2012). Social structure of facebook networks. Physica A: Statistical Mechanics and its Applications, 391(16), 4165-4180.

FILE FORMAT

- Data is stored as a text file. There is a separate file for each network.
- Instead of the actual names of the accounts, a unique integer identifier is given to each account. The identifiers for each file will always start from $0,1,2, \dots , n-1$, where n is the number of accounts in the network.
- The first line of the file contains two integers n and e, separated by a space. n is the number of accounts in the network, while e is the number of friendships in the network.
- For this project, your code must **at least** be able to handle up to 5000 accounts in a single network. In the example data provided, this includes: **Caltech36**, **Rice31**, and **Trinity100**.
- This is followed by e lines, each describing a link between two accounts. Each of these lines contains two integers a and b ($0 \leq a, b < n$). This means that a is friends with b.
- Remember that on Facebook, friendships are bi-directional. Therefore, if a is friends with b, it implies that b is also friends with a.
- Here is an example text file, and the corresponding social graph:

```
6 5
0 1
0 3
1 4
2 3
3 5
```



IMPLEMENTING THE PROGRAM

- For this major course output, your task is to write a **C or Java** program that provides the following functionalities:

1	Read a file in the format given above and load the graph into an appropriate data structure for processing.
2	Given an ID number, display that person's friend list.
3	Given two ID numbers, display the connection between those two people in the network, if it exists.

- You are allowed to use standard C or Java libraries for this project, but **not** external ones. Note that because C and Java both do not have built-in support for graphs, it is up to you to decide how to build your own graph representation using the existing libraries.
- While both C and Java are allowed, please be aware that in general, there are some algorithms that, while possible, are quite cumbersome to implement in C due to the lack of support for convenient data structures like dynamic lists (ArrayList). Therefore, please take this into consideration when selecting the language you want to use.

PART 1: LOADING THE GRAPH

- Your program should be able to load the data in the file format given above.
- To receive full credit, your program must accept as input the file name or file path of the text file (i.e., the file name should not be hardcoded).
- After reading the file, the program must store it in the appropriate data structure. Keep in mind the worst-case scenario in terms of the dataset sizes.

- The data structure and the reason it was used should be discussed clearly in the report (see below).
- After loading the graph, the program should display a menu to access the two other features below (display friend list, display connections). This menu should be displayed repeatedly until the user chooses to exit the program.

PART 2: DISPLAY FRIEND LIST

- If the user chooses this option, the program should ask for the ID number of a person.
- If the ID number does not exist in the dataset, the program should display some error message **without terminating**[1].
- Otherwise, the program should display the list of friends of that person, as well as the total number of friends of that person.

PART 3: DISPLAY CONNECTIONS

- If the user chooses this option, the program should ask for two ID numbers a and b.
- If a or b does not exist in the dataset, the program should display some error message **without terminating[1]**.
- Otherwise, the program should display the **connection** between a and b, if it exists.
  - We define a connection as follows: a **connection** is a sequence a, c_1, c_2, c_3 … , c_n, b such that a is friends with c_1, c_1 is friends with c_2, c_2 is friends with c_3, …, c_n is friends with b.
- If a connection does not exist, the program should state that a connection could not be found **without terminating[1]**.

- Please check the Appendix section for a sample run of the program.

WRITING THE REPORT

- You are to write a report containing, at the minimum, the following information.
  1. A brief introduction to the project and an outline of the contents of the report.
  2. A clear description of the data structure used in representing the social graph data, including the rationale behind the use of such data structure/s. **Show supporting code if necessary.**
  3. A clear description of the algorithms used to display the friend list as well as to display the connection between two people in the network.
  4. An algorithmic analysis of the algorithms in terms of time complexity.
  5. Summary of findings and the group's learnings, insights, and realizations **with respect to data structures and graphs** after accomplishing this project.

- The beginning of the report should contain the **list of members**, as well as the **group number**.

- At the end of the report, there should be a **table detailing the contributions of each group member**, as well as a **References** section if needed. References are to be cited using APA format.

[1] The program **must** not end/terminate at this point. It displays the menu and wait for the user's selection.

- There is no required format for the report. However, keep in mind the following:
  1. The report is graded based on substance, not the length or the number of pages. When writing, go directly to the point; avoid long blocks of text without substance.
  2. Make sure that you cover all the points enumerated above in a substantial manner; otherwise, you won't get a perfect grade.
  3. The report should have a clear narrative; its purpose is to convey your group's journey in doing the project and the learnings and insights you gained from it.

Working with Groupmates

For this project, you are encouraged to work in groups of at most 4 members. Make sure that each member of the group has approximately the same amount of contribution to the project. Problems with groupmates must be discussed internally within the group and, if needed, with your instructor.

All members of the group should have a clear understanding of the submitted project, regardless of the division of tasks. Collaboration should be done with a genuine concern for the outcome of the project – not just for the sake of distributing the workload. The instructor may ask **any member** of the group regarding the submitted project. Failure to demonstrate that the member has a substantial understanding of the submitted work will result in **major deductions**.

Academic Honesty Policy

Honesty policy applies. You should explicitly acknowledge the source, i.e., the author (if available) and the URL of any resources that you used for the making of the project. You can write this acknowledgment in the report.

The student handbook states that (Sec. 5.2.4.2):

"Faculty members have the right to demand the presentation of a student's ID, to give a grade of 0.0, and to deny admission to class of any student caught cheating under Sec. 5.3.1.1 to Sec. 5.3.1.1.6. The student should immediately be informed of his/her grade and barred from further attending his/her classes."

The student handbook also states that (Sec. 10.3):

A student caught cheating, as defined in Sec. 5.3.1.1., shall be penalized with a grade of 0.0 in the requirement or in the course, at the discretion of the faculty member, without prejudice to an administrative sanction. In cases of alleged cheating, the faculty member should report the incident to the Student Discipline Formation Office (SDFO).

Sample Run

```
Input file path: mysamplefile.txt
Graph loaded!
MAIN MENU
[1] Get friend list
[2] Get connection
[3] Exit

Enter your choice: 1
Enter ID of person: 3

Person 3 has 5 friends!
List of friends: 16 726 740 1744 2564

MAIN MENU
[1] Get friend list
[2] Get connection
[3] Exit

Enter your choice: 1
Enter ID of person: 740

Person 3 has 20 friends!
List of friends: 2 3 14 22 50 80 111 122 140 150 199 600 783 956 1035 1833 1834 1835 1990 1993

MAIN MENU
[1] Get friend list
[2] Get connection
[3] Exit

Enter your choice: 2
Enter ID of first person: 1111
Enter ID of second person: 1993
There is a connection from 1111 to 1993!
1111 is friends with 3
3 is friends with 740
740 is friends with 1993

MAIN MENU
[1] Get friend list
[2] Get connection
[3] Exit

Enter your choice: 2
Enter ID of first person: 2204
Enter ID of second person: 2205
Cannot find a connection between 2204 and 2205

MAIN MENU
[1] Get friend list
[2] Get connection
[3] Exit

 Enter your choice: 3
```

MCO2 RUBRIC

TASK	POINTS			WEIGHT
Reading the File	**8** File is read correctly, and the proper format is followed.	**4** File is read correctly but some aspects of the format are not followed.	**0** File is not being read correctly.	**8**
Graph Representation	**15** An appropriate data structure is used to represent the graph in the program, and the report justifies it well.	**10 OR 5** An appropriate data structure was used but there are minor implementation errors, or the report fails to clearly explain the data structure and why it is appropriate.	**0** There was a substantial lack of understanding of how the graph should be represented.	**15**
Program Flow	**12** The program flow complies with the requirements and does not crash.	**8 OR 4** The program flow does not fully comply with the requirements, or crashes / terminates unexpectedly on certain occasions.	**0** The program does not have a flow, i.e., the features are all separately implemented.	**12**
Implementation of Display Friend List	**11** The algorithm is correctly implemented and returns the correct results.	**8 OR 4** Mistakes in the implementation of the algorithm.	**0** There is a substantial misunderstanding of the algorithm.	**11**
Implementation of Display Connection	**20** The algorithm is correctly implemented and returns the correct results.	**15 OR 10 OR 5** Mistakes in the implementation of the algorithm.	**0** There is a substantial misunderstanding of the algorithm.	**20**
Report and Analysis of	**12**	**8 OR 4**	**0**	**12**

Display Friend List	The report clearly explains the Display Friends algorithm and provides a clear analysis of its implementation and time complexity.	The algorithm is discussed using vague, surface-level terms, or the implementation and analysis of the algorithm is not clearly discussed.	The algorithm is not discussed, or the discussion shows a substantial misunderstanding of the task.	
Report and Analysis of Display Connection	**12** The report clearly explains the Display Connections algorithm and provides a clear analysis of its implementation and time complexity.	**8 OR 4** The algorithm is discussed using vague, surface-level terms, or the implementation and analysis of the algorithm is not clearly discussed.	**10** The algorithm is not discussed, or the discussion shows a substantial misunderstanding of the task.	**12**
Summary and Insights	**5** Report summarizes the findings as well as the learnings and insights gained from the project.	**3** Report summarizes the findings but the learnings and insights obtained from the project are surface-level.	**0** Report does not summarize the findings or insights.	**5**
Overall Report Quality	**5** Report is clear, comprehensive, well-written, and has a good flow / narrative.	**3** Report is not well-written and is not cohesive / does not follow a flow or narrative.	**0** Report shows a substantial disregard to quality.	**5**