MIE 223 Data Science
Lab and Assignment 07:
Graph Data Analysis

In this lab and assignment, you will learn and gain hands-on experience working with and analyzing Graph/Network data.

- Programming language: Python (Google Colab Environment)
- Due Date: Posted in Syllabus

Marking scheme and requirements: You are required to use the Jupyter notebook named `Assignment_07.ipynb` to provide appropriate answers to the questions in this assignment. All changes to the `Assignment_07.ipynb` notebook should be committed and pushed to your assignment repository on GitHub.

This assignment has *4 points* in total and the point allocation is shown below:

- Solution to Questions (1 point):
  - Q1(a): 0.1 point
  - Q1(b): 0.1 point
  - Q1(c): 0.1 point
  - Q2(a): 0.1 point
  - Q2(b): 0.1 point
  - Q3(a): 0.1 point
  - Q3(b): 0.1 point
  - Q3(c): 0.1 point
  - Q4(a): 0.1 point
  - Q4(b): 0.1 point
- Code review (1 point)
- Quiz (2 points)

What/how to submit your work:

- All your code should be included in the provided notebook named `Assignment_07.ipynb`.
- Commit and push your work to your GitHub repository in order to submit it. Your last commit and push before the assignment deadline will be considered to be your submission. You can check your repository online to make sure that all required files have actually been committed and pushed to your repository.
- A link to create a personal repository for this assignment is posted on QUERCUS.

Notes that you should pay attention to

1. All your code should be written in the provided notebook named `Assignment_07.ipynb`.
2. All functions must *return* the specified return type.
3. Commit and push your work to your GitHub repository in order to submit it. **Your last commit and push will be considered to be your submission.** You can check your repository online to make sure that all required code has actually been committed and pushed to your repository.
4. **Your code should have output for all assignment questions unless otherwise stated**. If any of the assignment questions asks you to explain your outputs, please use markdown cells to clearly explain them.
5. As a backup plan, also upload a PDF copy of your assignment notebook to GitHub. This is useful especially when GitHub cannot render some images or outputs.
6. During the code review, TAs will go over your answer submissions to assess your understanding of the concepts relating to the assignment and the logic behind your codes. Please, ensure your codes are readable and well commented using python comments and markdown cells when necessary. Not have readable code may impact your code review score.
7. Please note the plagiarism policy in the syllabus.

1 Main Assignment

Please answer the questions below in the provided notebook named `Assignment_07.ipynb` using Google Colab. **NetworkX** functions should be used for all graph methods. **Clearly give titles and labels to all plots. Use a figure size of 15x10 for all plots.**

Dataset Description: The widely acclaimed TV series "Game of Thrones" by HBO, adapted from the equally popular book series "*A Song of Ice and Fire*" by George R.R. Martin, serves as the basis for this assignment. Our focus will be on the character co-occurrence network within the Game of Thrones books. In this context, two characters are said to co-occur if their names are found within 15 words of each other in the text. As a data scientist, you want to be able to analyze these graph data to uncover entities' relations, characters' importance or even story plots. Some questions you might want to consider are:

- How can we abstract the provided information into graphs for further analysis?
- What knowledge can we gain from the degree of nodes in this context?
- What can we learn about the story and characters from the centrality of nodes?
- How might we apply time-series analysis techniques from previous assignments to extract meaningful insights?

Using the **book1_df, book2_df, book3_df, book4_df,** and **book5_df** DataFrame, complete the following questions.

Q1. Graph Basics

(a) Construct an undirected graph **book1_G** by utilizing **book1_df**, treating it as an edge list. In this dataframe, the "Source" and "Target" columns contain character names, which should be utilized as **node labels**, representing the characters (therefore, list(book1_G.nodes()) should return a list of string character names). Each dataframe row denotes a connection, with the "Source" and "Target" columns indicating the linked characters, and the "Weight" column specifying the edge's weight attribute. Finally, output the total count of nodes and edges in the book1 graph.

(b) For each of the five books, replicate the process described in (a) to create graphs named **book2_G**, **book3_G**, **book4_G**, and **book5_G**. For each of these graphs (5 of them), display the top-3 edges based on the highest weight attribute.

(c) Construct a single undirected graph, **allbook_G**, which encompasses all the edges from the five books. In this graph, the weight attribute of each edge should be the cumulative total from all five books (consider the method for this, recalling from the lab that merely adding new nodes/edges may overlook existing ones). Identify and list the top-5 edges in **allbook_G** based on their cumulative weight attribute.

Q2. Visualization

(a) Create a visualization of **book1_G** using the spring layout. In this plot, adjust the parameter in 'nx.draw()' to reveal node labels, keeping all other settings like node_size and font_color at their defaults, except 'ax=ax' if needed.

(b) Create a visualization of **book1_G** using the other two layouts we explored in the lab: spectral layout and circular layout. In this plot, adjust the parameter in 'nx.draw()' to reveal node

labels, keeping all other settings like node_size and font_color at their defaults, except 'ax=ax' if needed.

Q3. Centrality

(a) Determine the degree centrality for **allbook_G** and list the top 15 nodes with the highest centrality values. Assess whether these characters are major or minor characters.

(b) Perform the same analysis as in (a) for PageRank, Closeness Centrality, and Betweenness Centrality.

(c) Apply the 'draw_centrality()' function from Lab-7 to create a circular layout visualization representing Closeness Centrality for **book1_G**. Set the 'node_scale' parameter to 1e3 (1e3 = 10^3).

Q4. Analysis

(a) *A Song of Ice and Fire* employs a unique writing style known as Point of View (PoV), where each chapter offers the perspective of a specific character, sharing their thoughts and shaping the storyline. George R. R. Martin's third-person narrative style ensures that the name of the PoV character is consistently mentioned, even when the story is being told from their perspective. Typically, each book in the series includes about 8 to 12 PoV characters. Employ betweenness centrality as a metric to identify the top-5 likely PoV characters in each book.

(b) Utilizing the top-15 characters identified in question 3(a), create a "time-series" graph showing the degree centrality of each character across all five books, sequenced by book number. Assign a degree centrality of 0 for characters not present in a particular book. From this analysis, identify one character whose importance escalates up to book 4, and another character whose significance diminishes throughout the series.