**Laboratory Session 3**

Manipulation using NetworkX and other python libraries for Community detections

A)

Use the karateClubDataset used in Laboratory session 1 or downloaded elsewhere https://docs.dgl.ai/en/2.0.x/generated/dgl.data.KarateClubDataset.html

1) Study and execute the program below for community detection and display the plot showing the communities detected from KarateClubDataset and save the result as variables. Suggest alternative visualization that enhances readability of the communities (various color and thickness or annotation).

```
import matplotlib.pyplot as plt
import networkx as nx
from networkx.algorithms.community.centrality import girvan_newman

G = nx.karate_club_graph()
communities = girvan_newman(G)

node_groups = []
for com in next(communities):
  node_groups.append(list(com))

print(node_groups)

color_map = []
for node in G:
    if node in node_groups[0]:
        color_map.append('blue')
    else:
        color_map.append('green')
nx.draw(G, node_color=color_map, with_labels=True)
plt.show()
```

2) We want to study the result of community detection using Ratio Cut Method. For this purpose, use the inbuilt function in NetworkX `from community.kernighan_lin_bisection()` and display the result of the community detection, and save the result in a separate variables.

3) Repeat 2) when using the Louvain community detection algorithm, which has inbuilt function in NetworkX

4) Repeat 2) when using label-propagation community detection algorithm.

5) We want to evaluate the quality of the communities detected in 1)-4). For this purpose, study the available community quality performance algorithms inbuilt in NetworkX and write a script that outputs the quality performance (s) for each of the detection algorithm (Girvan Neuman, Ratio Cut, label propagation and Louvain method).

6) We want to compare the performance of the algorithms in 1)-4) in terms of algorithmic complexity. For this purpose, write a script that reports the execution time of each algortihm (Girvan Neuman, Ratio Cut, label propagation and Louvan method).

7) We want to seek a random graph that has close characteristic to Karate graph in terms of communities. For this purpose, use the function gnm_random_graph ( ) where n corresponds to the total number of nodes of Karate graph and m is the total number of edges of the Karate graph. Write a script that generates various random graph by varying the seed value of the gnm_random_graph function and determine for each graph its corresponding communities using one of the algorithms studied above. Determine the random graph that is close enough in terms of number of partition to Karate dataset.

8) We want to repeat 1)-6) when using overlapping communities. For this purpose, explore the dataset repositories of graph dataset provided in course handout and suggest an example of large scale overlapping graph dataset of your choice and repeat the steps 1-6).