DATA SCIENCE MASTER 2023-2024
MADE BY : LAGHDAF KHAOULA AND BELLAKHDAR OMAIMA
SUPERVISED BY : Mr.TARIK AGOUTI

PROJECT REPORT:
COMMUNITIES DETECTION IN COMPLEX NETWORKS

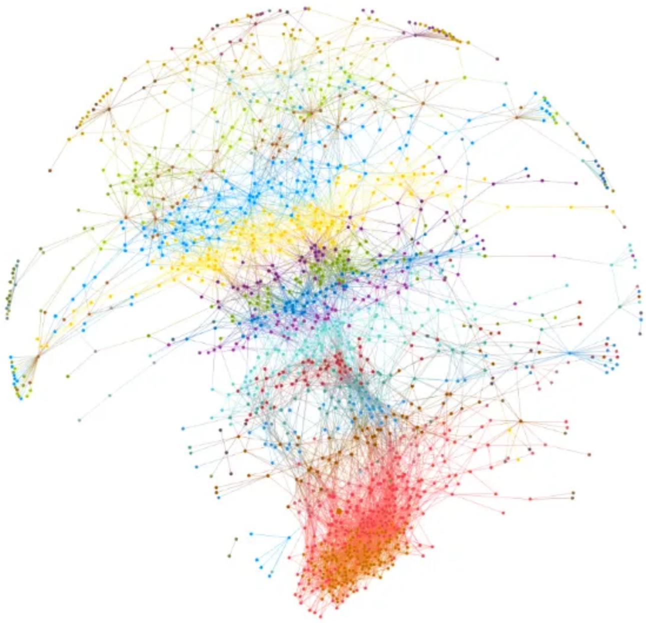


TABLE OF CONTENT :

I. INTRODUCTION :

The ever-growing interconnectedness of individuals and entities in various systems has led to the emergence of complex networks, where relationships and interactions play a crucial role in understanding the structure and dynamics of these networks. One essential aspect of analyzing complex networks is the identification of communities, or groups of nodes that exhibit higher internal connectivity compared to the rest of the network.

This project report focuses on the application of advanced algorithms for detecting communities within complex networks, with a particular emphasis on the renowned **Zachary Karate Club** dataset.
Applying the communities algorithms on Zachary Karate Club dataset should be answering the following questions :

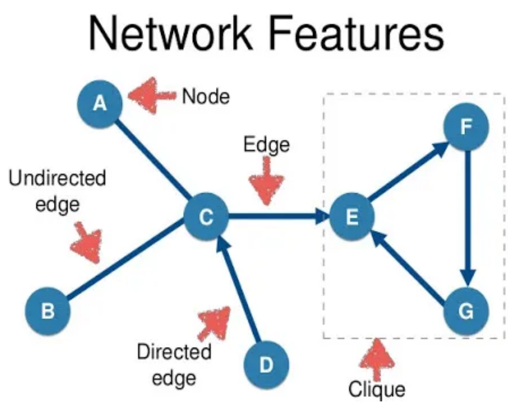
- ➔ How does the network of the karate club naturally partition into communities or groups?
- ➔ Which members are more closely connected with each other, forming distinct subgroups within the club?
- ➔ Are there specific subgroups within the karate club that share stronger connections among themselves than with the rest of the club?
- ➔ What is the most influential node in the karate club network?

Through this exploration, the report intends to contribute insights into the community structure of the Zachary Karate Club network. Applying the **FAST GREEDY** and **LABEL-PROPAGATION** algorithms will shed light on potential patterns of influence, collaboration, or conflict among its members.

II. MAIN TERMS OF COMPLEX NETWORKS:

Complex Network studies how to recognise, describe, visualize and analyze complex networks. The most prominent way of analyzing networks is using Python Library **NetworkX** which provides a prominent way for constructing and drawing complex Neural Networks.

STRUCTURE :
The two most important concepts concerning networks are entities and the relationships between them. Entities are referred to as **nodes** and relationships are known as **edges.** When it is essential, we represent **nodes** and **edges** by adding properties also known as attributes. A relationship or edge typically involves two discrete entities or nodes although an entity can be in a relationship with itself, such a relationship is referred to as **reflexive.**



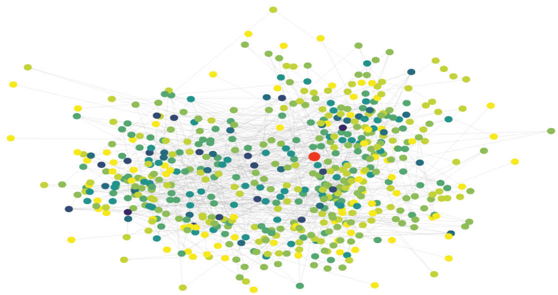
→ Large graphs of real life are called complex networks.

2.1 COMMUNITIES:

The concept of communities in complex networks refers to the identification of strongly interconnected groups of nodes within a network. Communities represent subsets of nodes that are more densely connected to each other than to the rest of the network.

2.2 INFLUENTIAL NODES:

Influential nodes are nodes that lead to faster and wider spreading in complex networks. To identify them we often use measures such as **degree centrality DC**, **betweenness centrality BC**, and **closeness centrality CC**.



➔ Degree Centrality:

Definition: Degree centrality is a measure of how many connections a node has.
Calculation: It is simply the number of edges connected to a node.
Influence: Nodes with high degree centrality are often considered influential because they have a large number of direct connections.

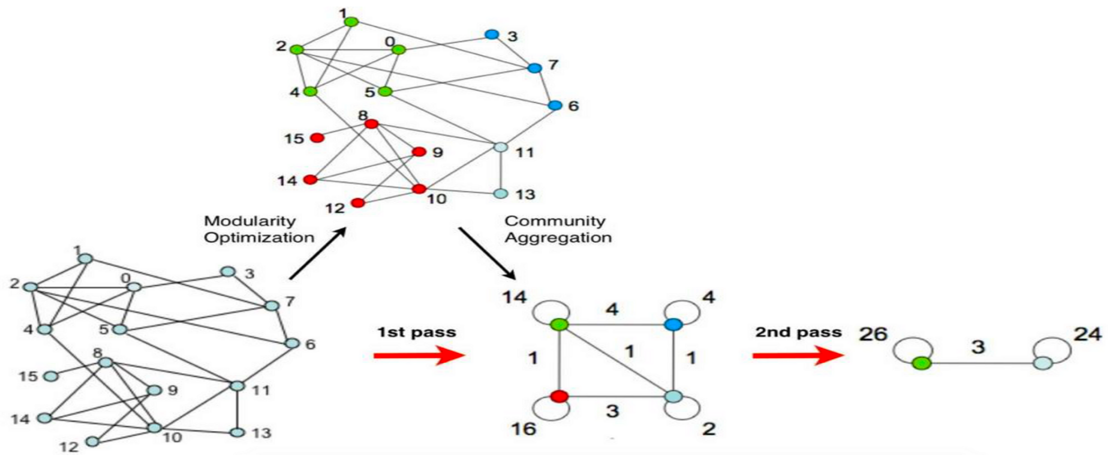
➔ Betweenness Centrality:

Definition: Betweenness centrality identifies nodes that act as bridges between different parts of a network.
Calculation: It measures the number of shortest paths that pass through a node.
Influence: Nodes with high betweenness centrality can control the flow of information between other nodes.

➔ Closeness Centrality:

Definition: Closeness centrality is a measure that identifies nodes in a network based on how quickly they can reach other nodes.
Calculation: It calculates the average length of the shortest paths from a node to all other nodes in the network.

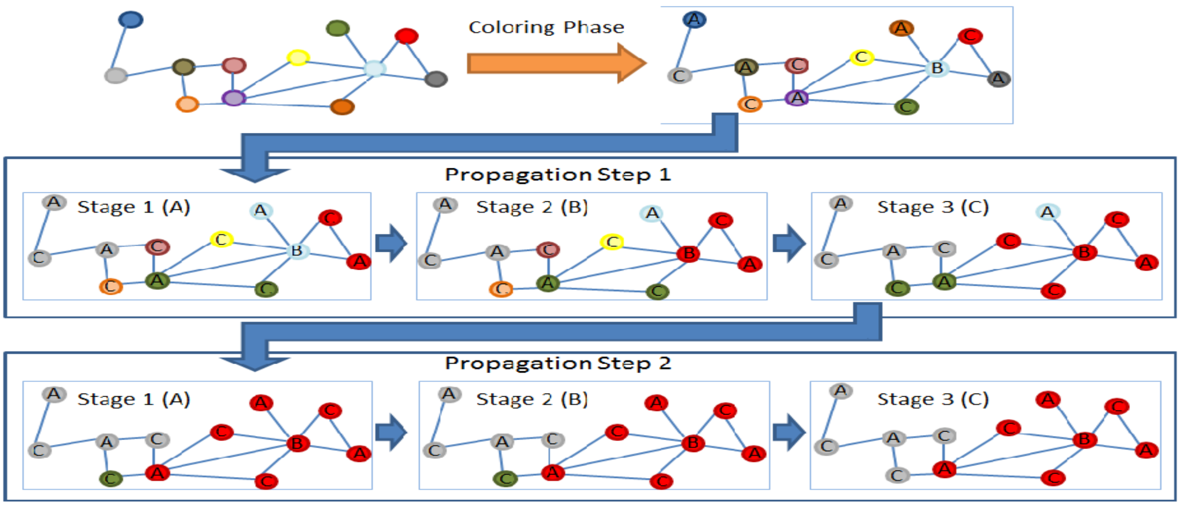
III. FAST GREEDY ALGORITHM:



The Fast Greedy algorithm is a community detection method aimed at finding clusters in a network by iteratively merging communities to maximize modularity. Summary:

- ➔ **Objective:** Maximize modularity by iteratively merging communities.
- ➔ **Agglomerative Approach:** Starts with individual nodes as separate communities and merges them for modularity improvement.
- ➔ **Edge-Based Optimization:** Evaluates impact of merging communities based on edges between nodes.
- ➔ **Greedy Strategy:** Iteratively merges the pair of communities that maximizes modularity at each step.
- ➔ **Fast Execution:** Designed for efficiency, particularly for large networks.
- ➔ **Modularity Definition:** Measures the quality of a network division into communities.
- ➔ **Iterative Steps:** Continues merging communities until no further increase in modularity can be achieved.
- ➔ **Output:** Produces a partition of nodes into communities that optimizes network modularity.

IV. LABEL PROPAGATION ALGORITHM:



Label Propagation is a community detection algorithm that identifies clusters in a network through iterative label updates. Summary:

- **Initialization:** Assign a unique label to each node.

- **Label Propagation:** Nodes update their labels to match the majority label among their neighbors in an iterative process.

- **Community Formation:** Nodes with the same label after iterations belong to the same community.

- **Dynamic Nature:** Labels evolve dynamically, making it suitable for networks with soft or overlapping communities.

- **Scalability:** Known for its efficiency, making it scalable for large networks.

- **Randomness:** In case of ties, randomness may be introduced in label assignments.

- **Heuristic Approach:** Simple and lightweight, doesn't rely on optimizing an explicit objective function.

- **Parameters:** May have parameters like maximum iterations or a convergence threshold.

V. THE DATASET BACKGROUND:

ZACHARY KARATE CLUB's Dataset, these are data collected from the members of a university karate club by Wayne Zachary.

- **Data Collection:**

Wayne Zachary collected data from members of a university karate club.
Two matrices are mentioned: ZACHE and ZACHC.
ZACHE **Matrix:** Represents the presence or absence of ties (connections or relationships) among the members of the karate club.
ZACHC **Matrix:** Indicates the relative strength of the associations, specifically the number of situations in and outside the club in which interactions occurred.

- **Purpose of the Study:**

Wayne Zachary used these collected data to conduct a study on the dynamics of the karate club.
The mention of an "information flow model of network conflict resolution" suggests that Zachary was interested in understanding how information and communication flow within the network and how conflicts are resolved.

- **Outcome of the Study:**

The statement indicates that Zachary used the collected data and the information flow model to explain the split-up of the karate club.
This split presumably occurred as a result of disputes among the members.

- **Reference:**

The information provided comes from a study conducted by Wayne Zachary in 1977.

In summary, Wayne Zachary conducted a social network analysis on a university karate club using matrices to represent the presence or absence of ties and the strength of associations among members. The study aimed to use an information flow model to explain the club's split following disputes among its members.

VI. THE TOOLS USED:

NetworkX is a Python library for the creation, analysis, and manipulation of complex networks or graphs. It provides tools for studying the structure and dynamics of networks and is widely used in various scientific disciplines, including social science, biology, physics, and computer science.

Matplotlib is a comprehensive 2D plotting library for the Python programming language. It provides a wide variety of static, animated, and interactive plots, charts, and visualizations. Matplotlib is widely used in scientific computing, data analysis, and other fields to create high-quality and publication-ready graphics.

Google Colab, short for Colaboratory, is a cloud-based platform provided by Google that allows users to write and execute Python code in a web-based interactive environment. It is particularly popular among data scientists, machine learning researchers, and educators for its convenience and accessibility.

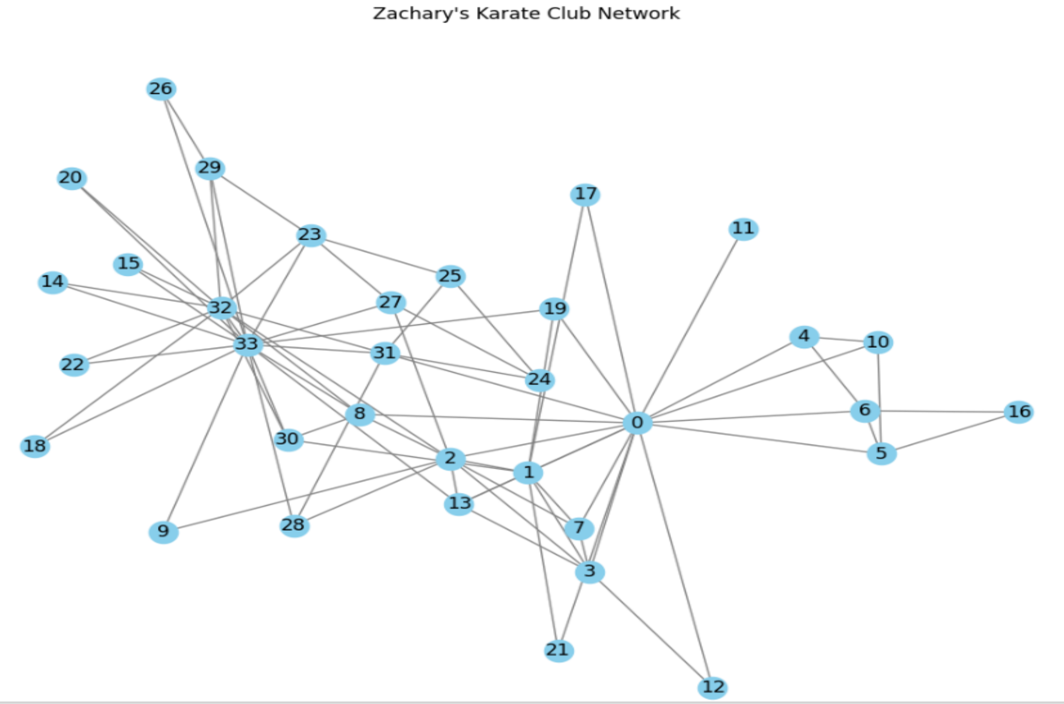
VII. APPLICATION:

➔ IMPORTATION OF ZACHARY KARATE CLUB NETWORK DATASET :

```python
import networkx as nx
import community
import matplotlib.pyplot as plt

# IMPORTATION OF Zachary's Karate Club network
G = nx.karate_club_graph()
# GRAPH PROPERTIES
print("Number of nodes:", G.number_of_nodes())
print("Number of edges:", G.number_of_edges())

# GRAPH VISUALIZATION
pos = nx.spring_layout(G)
plt.figure(figsize=(10, 8))
nx.draw(G, pos, with_labels=True, node_size=300, node_color="skyblue", edge_color="gray")
plt.title("Zachary's Karate Club Network")
plt.show()
```

➢ **Output:** The Graph Of Data

```
Number of nodes:  34
Number of edges:  78
```



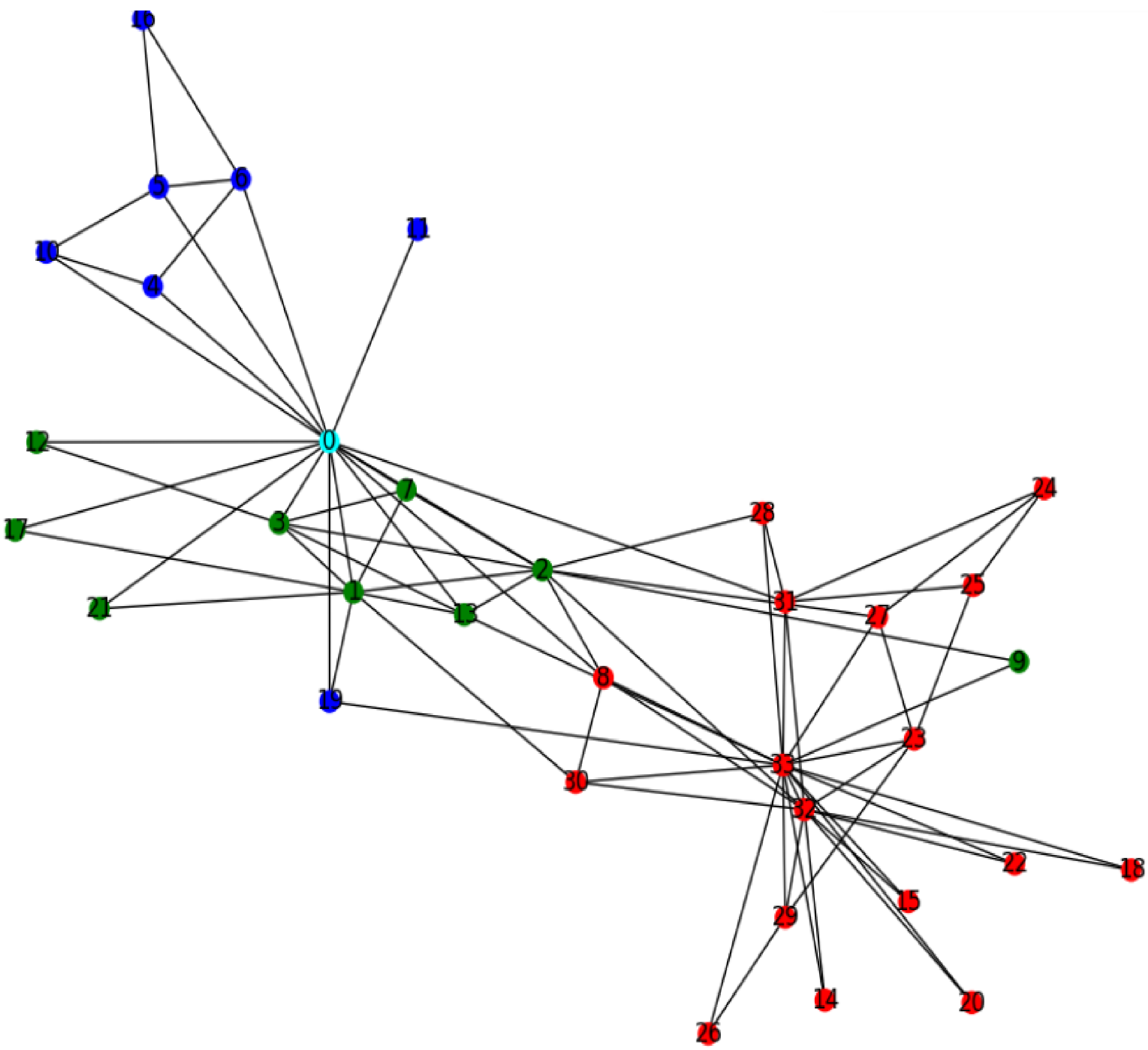
→ APPLICATION OF **FAST GREEDY** ALGORITHM ON ZACHARY DATA, WITH IDENTIFYING THE MOST INFLUENTIAL NODE:

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


# For illustration, we'll use the Zachary's Karate Club graph
G = nx.karate_club_graph()

# Convert the graph to an undirected graph
G = G.to_undirected()

# Apply Fast Greedy algorithm for community detection
communities = list(community.greedy_modularity_communities(G))

# Create a mapping of nodes to community IDs
community_mapping = {node: i for i, comm in enumerate(communities) for node in comm}

# Define colors for communities
community_colors = {
    0: 'red',
    1: 'green',
    2: 'blue',
    # Add more colors if needed
}
```

```
# Get node colors based on community mapping
node_colors = [community_colors[community_mapping[node]] for node in G.nodes]

# Calculate betweenness centrality
betweenness_centrality = centrality.betweenness_centrality(G)

# Identify the most influential nodes (highest betweenness centrality)
most_influential_nodes = max(betweenness_centrality, key=betweenness_centrality.get)

# Add color cyan to the most influential nodes
node_colors[most_influential_nodes] = 'cyan'

# Draw the graph with nodes colored by communities and highlight most influential nodes in cy
pos = nx.spring_layout(G)
plt.figure(figsize=(10, 8))
nx.draw(G, pos, node_size=100, node_color=node_colors, with_labels=True)
plt.title("Our Network with Fast Greedy Communities and Most Influential Nodes")
plt.show()
```

➢ **Output:**

Our Network with Fast Greedy Communities and Most Influential Nodes



➢ **Analysis:**

The Fast Greedy algorithm detected a network structure with three distinct communities (blue, red, green) and identified an influential node (Node 0) highlighted in cyan. Here's the analysis:

Community Structure:
Blue Community: Represents a cohesive subgroup with strong internal connections.
Red Community: Another distinct group of nodes with dense internal connections.
Green Community: Comprises nodes with tight internal relationships.

Influential Node (Node 0):
Cyan Color: Node 0, highlighted in cyan, has high betweenness centrality.
Acts as a bridge between different communities, controlling the flow of information.

Modularity Optimization:
Fast Greedy successfully optimized modularity by partitioning the network into cohesive groups with limited connections between them.

➔ ***Interpretation:***

The detected communities provide insights into the social structure of the Zachary's Karate Club, revealing potential cliques or subgroups within the club.
The influential node suggests that certain individuals may act as mediators or central figures connecting different social circles within the club.

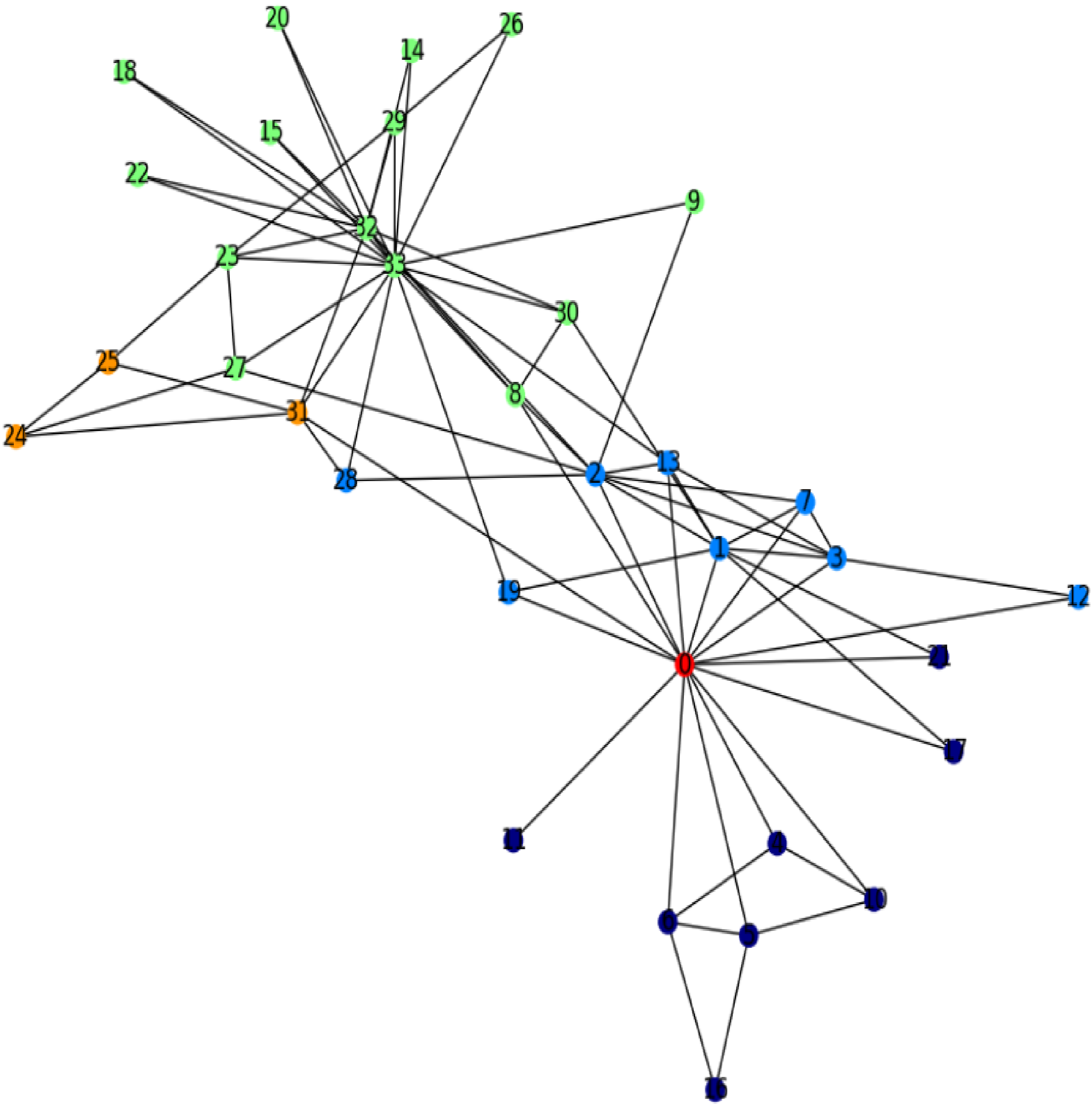
→ APPLICATION OF **LABEL PROPAGATION** ALGORITHM, WITH IDENTIFYING THE MOST INFLUENTIAL NODE:

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


# For illustration, we'll use the Zachary's Karate Club graph
G = nx.karate_club_graph()

# Convert the graph to an undirected graph
G = G.to_undirected()

# Apply Label Propagation algorithm for community detection
communities = list(label_propagation.asyn_lpa_communities(G))

# Create a mapping of nodes to dynamically assigned community colors
community_mapping = {node: i for i, comm in enumerate(communities) for node in comm}
num_communities = len(communities)

# Dynamically assign colors to communities
community_colors = {i: plt.cm.jet(i / num_communities) for i in range(num_communities)}

# Get node colors based on dynamically assigned community mapping
node_colors = [community_colors[community_mapping[node]] for node in G.nodes]
```

```
# Calculate betweenness centrality
betweenness_centrality = centrality.betweenness_centrality(G)

# Identify the most influential nodes (highest betweenness centrality)
most_influential_nodes = max(betweenness_centrality, key=betweenness_centrality.get)

# Add color cyan to the most influential nodes
node_colors[most_influential_nodes] = 'red'

# Draw the graph with nodes colored by communities and highlight most influential nodes in cy
pos = nx.spring_layout(G)
plt.figure(figsize=(10, 8))
nx.draw(G, pos, node_size=100, node_color=node_colors, with_labels=True)
plt.title("Your Network with Label Propagation Communities and Most Influential Nodes")
plt.show()
```

➢ **Output :**

Our Network with Label Propagation Communities and Most Influential Nodes



➢ **Analysis:**

The Label Propagation algorithm detected a network structure with four distinct communities (light blue, orange, light green, dark blue) and identified an influential node (Node 0) highlighted in red. Here's the analysis:

Community Structure:
-Light Blue Community: Represents a distinct subgroup with strong internal connections.
-Orange Community: Another cohesive group of nodes with dense internal relationships. The separation of the orange community from the light blue community indicates modularity in the network.
-Light Green Community: Reflects a subgroup of nodes with strong internal ties. The presence of multiple communities suggests a nuanced social structure.
-Dark Blue Community: Comprises nodes with tight internal connections, forming yet another cohesive group within the network.

Influential Node (Node 0):
Red Color (Node 0): Node 0 retains its influential status, suggesting its importance in connecting different communities. The cyan color highlights its role as a bridge between diverse subgroups.

Modularity Optimization:
The Label Propagation Algorithm has identified four communities, optimizing modularity by recognizing distinct sets of nodes with stronger internal ties.

Interpretation:
The communities detected by Label Propagation reveal finer-grained substructures within the Zachary's Karate Club, providing a more detailed view of social affiliations.
The influential node (Node 0) continues to act as a connector, maintaining communication and relationships between the identified communities.
The nuanced division into four communities suggests a more intricate social landscape, potentially reflecting different roles or affiliations within the karate club.

VIII. CONCLUSION:

In conclusion, this project has delved into the intricate realm of complex networks, specifically focusing on the detection of communities within the Zachary Karate Club dataset. Through the application of advanced algorithms such as FAST GREEDY and LABEL PROPAGATION, we have endeavored to unveil the hidden structures and inherent patterns of social interactions within the karate club.

The analysis has provided valuable insights into the community structure, revealing cohesive subgroups that reflect the underlying dynamics of relationships among the club members. By leveraging these algorithms, we have identified clusters that may signify alliances, fractures, or influential nodes within the network, contributing to a deeper understanding of the social fabric within the Zachary Karate Club.

In essence, this project not only demonstrates the applicability of sophisticated algorithms in uncovering hidden structures within complex networks but also underscores the importance of community detection as a powerful tool for deciphering the intricacies of social relationships.

IX. REFERENCES:

https://networkrepository.com/graphvis.php

http://konect.cc/networks/

https://neo4j.com/docs/graph-data-science/current/algorithms/label-propagation/

https://www.simplilearn.com/tutorials/data-structure-tutorial/greedy-algorithm