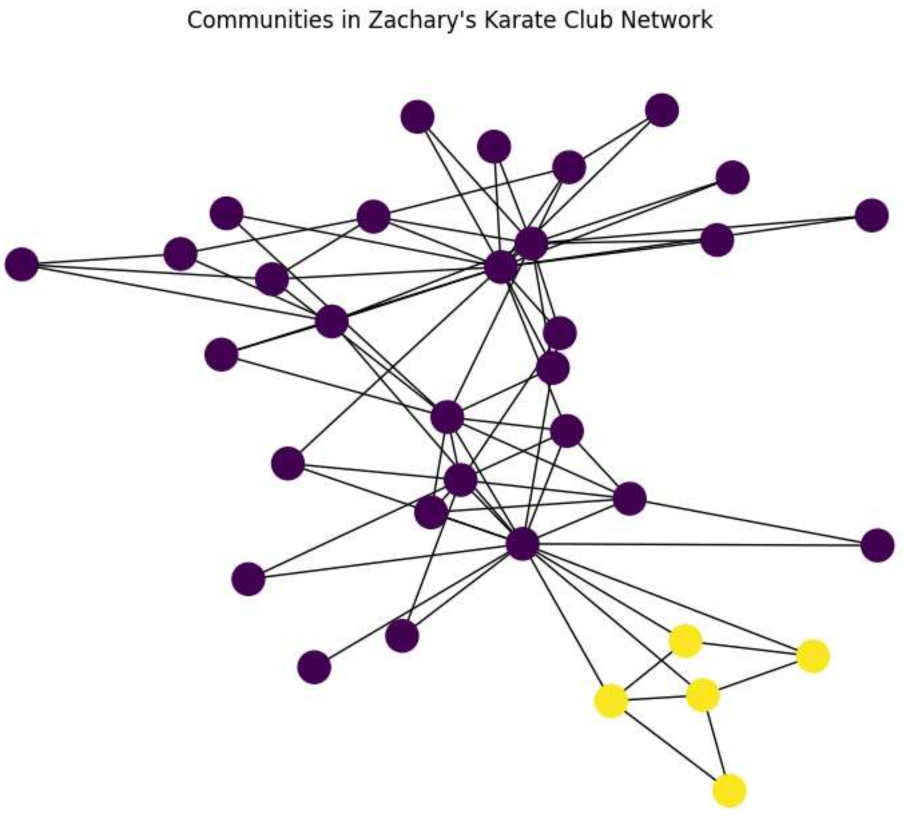
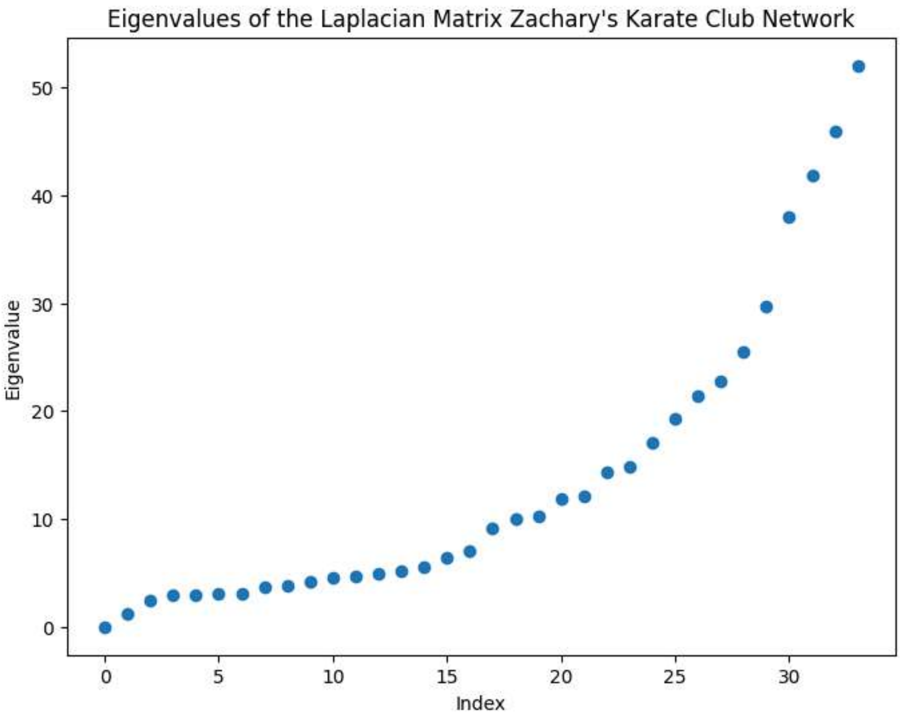
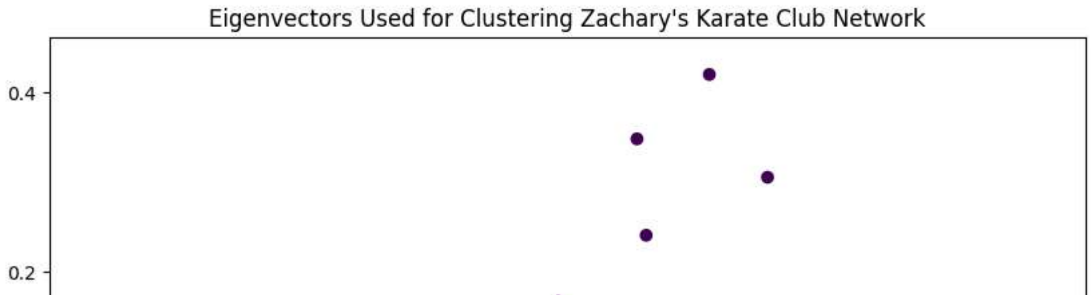
```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt

# Load the graph dataset (Zachary's Karate Club Network)
G = nx.karate_club_graph()

# Convert the graph to an adjacency matrix
A = nx.to_numpy_array(G)

# Compute the Laplacian matrix
D = np.diag(np.sum(A, axis=1))
L = D - A

# Compute the eigenvalues and eigenvectors of the Laplacian
eigenvalues, eigenvectors = np.linalg.eigh(L)

# Sort the eigenvalues and eigenvectors in ascending order
idx = np.argsort(eigenvalues)
eigenvalues = eigenvalues[idx]
eigenvectors = eigenvectors[:, idx]

# Select the eigenvectors corresponding to the smallest non-zero eigenvalues
k = 2  # Number of clusters
X = eigenvectors[:, 1:k+1]

# Perform k-means clustering on the eigenvectors
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=k, random_state=0).fit(X)
labels = kmeans.labels_

# Visualize the communities
pos = nx.spring_layout(G)
plt.figure(figsize=(10, 8))
plt.axis('off')
nx.draw_networkx_nodes(G, pos, node_color=labels, cmap='viridis')
nx.draw_networkx_edges(G, pos)
plt.title('Communities in Zachary\'s Karate Club Network')
plt.show()

# Plot the eigenvalues of the Laplacian matrix
plt.figure(figsize=(8, 6))
plt.scatter(range(len(eigenvalues)), eigenvalues)
plt.title('Eigenvalues of the Laplacian Matrix Zachary\'s Karate Club Network')
plt.xlabel('Index')
plt.ylabel('Eigenvalue')
plt.show()

# Plot the eigenvectors used for clustering
plt.figure(figsize=(10, 8))
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis')
plt.title('Eigenvectors Used for Clustering Zachary\'s Karate Club Network')
plt.xlabel('First Eigenvector')
plt.ylabel('Second Eigenvector')
plt.show()

# Load another dataset (Les Miserables co-appearance network)
G = nx.les_miserables_graph()

# Convert the graph to an adjacency matrix
A = nx.to_numpy_array(G)

# Compute the Laplacian matrix
D = np.diag(np.sum(A, axis=1))
L = D - A

# Compute the eigenvalues and eigenvectors of the Laplacian
eigenvalues, eigenvectors = np.linalg.eigh(L)

# Sort the eigenvalues and eigenvectors in ascending order
idx = np.argsort(eigenvalues)
eigenvalues = eigenvalues[idx]
eigenvectors = eigenvectors[:, idx]

# Select the eigenvectors corresponding to the smallest non-zero eigenvalues
k = 2  # Number of clusters
```

```
    X = eigenvectors[:, 1:k+1]

    # Perform k-means clustering on the eigenvectors
    kmeans = KMeans(n_clusters=k, random_state=0).fit(X)
    labels = kmeans.labels_

    # Visualize the communities
    pos = nx.spring_layout(G)
    plt.figure(figsize=(10, 8))
    plt.axis('off')
    nx.draw_networkx_nodes(G, pos, node_color=labels, cmap='viridis')
    nx.draw_networkx_edges(G, pos)
    plt.title('Communities in Les Miserables Co-appearance Network')
    plt.show()

    # Additional visualizations for Les Miserables Network
    degree_sequence = sorted([d for n, d in G.degree()], reverse=True)
    plt.figure(figsize=(8, 6))
    plt.loglog(degree_sequence, 'b-', marker='o')
    plt.title('Degree Rank Plot Les Miserables Network')
    plt.xlabel('Rank')
    plt.ylabel('Degree')
    plt.show()

    clustering = nx.clustering(G)
    plt.figure(figsize=(8, 6))
    plt.scatter(list(clustering.values()), list(clustering.values()))
    plt.title('Node Clustering Coefficients Les Miserables Network')
    plt.xlabel('Node')
    plt.ylabel('Clustering Coefficient')
    plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10
  warnings.warn(
```


Communities in Zachary's Karate Club Network


Eigenvalues of the Laplacian Matrix Zachary's Karate Club Network


Eigenvectors Used for Clustering Zachary's Karate Club Network