Assignment-05-Ranjit-Menon

December 27, 2023

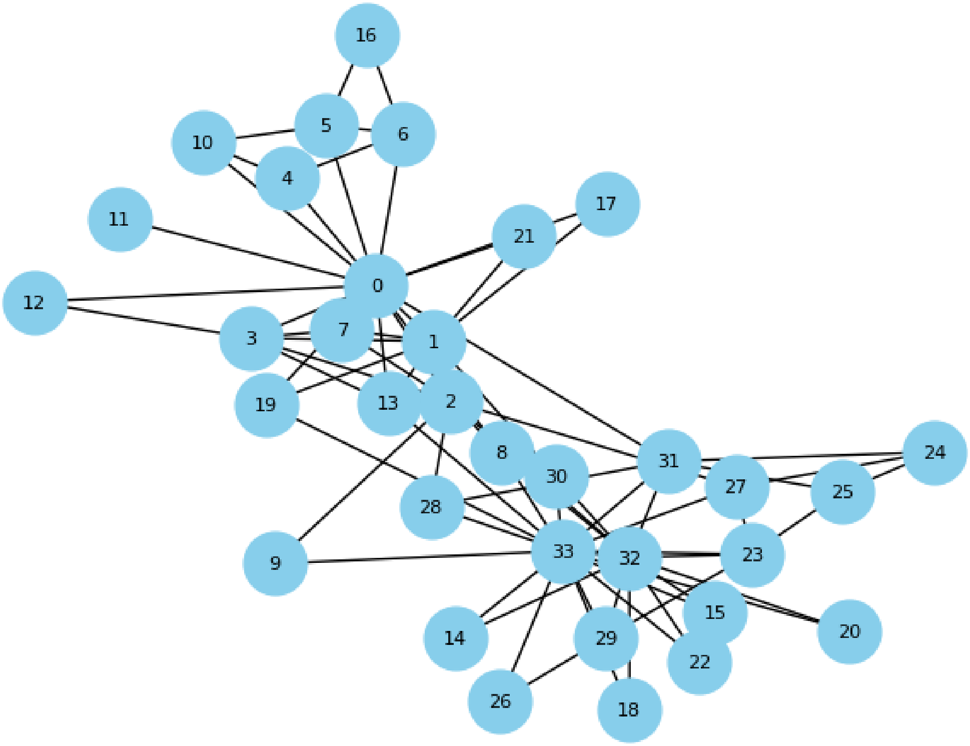
##

Complex NetworkAnalysis - Assignment 5

```
[1]: import networkx as nx
     import matplotlib.pyplot as plt
```

0.0.1 1. Visualize Karate club graph

```
[2]: G = nx.karate_club_graph()
     pos = nx.spring_layout(G)
     nx.draw(G, pos, with_labels=True, node_size=700, node_color='skyblue',␣
      ↪font_size=8)
```



0.0.2 2. Find the edge betweenness of each edge of the karate club graph

Helper method to run the code multiple times for finding edge and removing the nodes with max betweenness, another method will visualize the nodes based on partition.

[3]:
```
#method to calculate the edge betweeness
def calculate_edge_betweenness(graph):
    edge_betweenness = nx.edge_betweenness_centrality(graph)

    # iterate and print the edge betweeness centrality for each edges
    for edge, betweenness in edge_betweenness.items():
        print(f"Edge {edge}: Betweenness Centrality = {betweenness}")

#method to draw the graph with partition, this will dispaly community nodes␣
 ↪with different color
def draw_graph_with_partition(graph):

    # Detects communities in the graph using label propagation algorithm.
    communities = list(nx.community.label_propagation.
 ↪label_propagation_communities(graph))

    print(communities)
    node_colors = {}
    for i, community in enumerate(communities):
        for node in community:
            node_colors[node] = i

    # Visualize the partitioned graph
    pos = nx.spring_layout(graph)
    plt.figure(figsize=(8, 8))
    nx.draw(graph, pos, with_labels=True, node_size=700,␣
 ↪node_color=[node_colors[node] for node in graph.nodes()], cmap=plt.cm.tab10)
    plt.show()

#method to remove the edge with high betweeness
def remove_edge_with_high_betweenness(graph):

    # Calculate edge betweenness centrality
    edge_betweenness = nx.edge_betweenness_centrality(graph)
    max_betweenness_edge = max(edge_betweenness, key=edge_betweenness.get)

    # Print information about the edge to be removed
    print(f"Removing edge {max_betweenness_edge} with highest betweenness␣
 ↪centrality")
```

```
    # we will unpack (*) the tuple and pass this to remove_edge method
    graph.remove_edge(*max_betweenness_edge)
```

0.0.3 2.a. Calculate the edge betweenness centrality for every edge for the Karate club graph displayed above

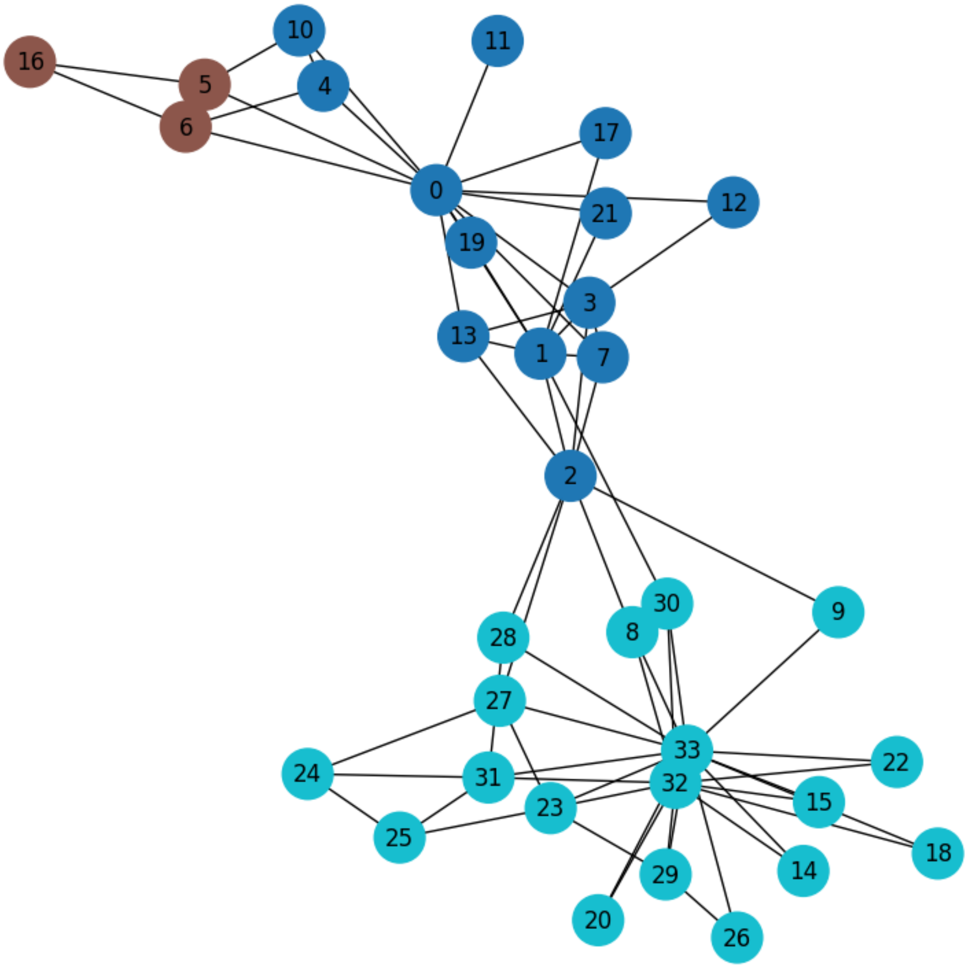
[4]:
```
calculate_edge_betweenness(G)
```

```
Edge (0, 1): Betweenness Centrality = 0.025252525252525245
Edge (0, 2): Betweenness Centrality = 0.0777876807288572
Edge (0, 3): Betweenness Centrality = 0.020499108734402855
Edge (0, 4): Betweenness Centrality = 0.0522875816993464
Edge (0, 5): Betweenness Centrality = 0.07813428401663694
Edge (0, 6): Betweenness Centrality = 0.07813428401663695
Edge (0, 7): Betweenness Centrality = 0.0228206434088787
Edge (0, 8): Betweenness Centrality = 0.07423959482783014
Edge (0, 10): Betweenness Centrality = 0.0522875816993464
Edge (0, 11): Betweenness Centrality = 0.058823529411764705
Edge (0, 12): Betweenness Centrality = 0.04652406417112298
Edge (0, 13): Betweenness Centrality = 0.04237189825425121
Edge (0, 17): Betweenness Centrality = 0.04012392835922248
Edge (0, 19): Betweenness Centrality = 0.045936960642843
Edge (0, 21): Betweenness Centrality = 0.040123928359222474
Edge (0, 31): Betweenness Centrality = 0.1272599949070537
Edge (1, 2): Betweenness Centrality = 0.023232323232323233
Edge (1, 3): Betweenness Centrality = 0.0077243018419489
Edge (1, 7): Betweenness Centrality = 0.007422969187675069
Edge (1, 13): Betweenness Centrality = 0.01240556828792123
Edge (1, 17): Betweenness Centrality = 0.01869960105254222
Edge (1, 19): Betweenness Centrality = 0.014633732280791102
Edge (1, 21): Betweenness Centrality = 0.01869960105254222
Edge (1, 30): Betweenness Centrality = 0.032280791104320514
Edge (2, 3): Betweenness Centrality = 0.022430184194890075
Edge (2, 7): Betweenness Centrality = 0.025214328155504617
Edge (2, 8): Betweenness Centrality = 0.009175791528732704
Edge (2, 9): Betweenness Centrality = 0.030803836686189627
Edge (2, 13): Betweenness Centrality = 0.007630931160342923
Edge (2, 27): Betweenness Centrality = 0.04119203236850296
Edge (2, 28): Betweenness Centrality = 0.02278244631185807
Edge (2, 32): Betweenness Centrality = 0.06898678663384543
Edge (3, 7): Betweenness Centrality = 0.003365588659706307
Edge (3, 12): Betweenness Centrality = 0.012299465240641705
Edge (3, 13): Betweenness Centrality = 0.01492233256939139
Edge (4, 6): Betweenness Centrality = 0.0047534165181224
Edge (4, 10): Betweenness Centrality = 0.0029708853238265
Edge (5, 6): Betweenness Centrality = 0.0029708853238265003
Edge (5, 10): Betweenness Centrality = 0.0047534165181224
Edge (5, 16): Betweenness Centrality = 0.029411764705882353
Edge (6, 16): Betweenness Centrality = 0.029411764705882353
Edge (8, 30): Betweenness Centrality = 0.00980392156862745
Edge (8, 32): Betweenness Centrality = 0.0304416716181422
Edge (8, 33): Betweenness Centrality = 0.04043657867187279
Edge (9, 33): Betweenness Centrality = 0.029615482556659026
Edge (13, 33): Betweenness Centrality = 0.06782389723566191
Edge (14, 32): Betweenness Centrality = 0.024083977025153497
Edge (14, 33): Betweenness Centrality = 0.03473955238661121
Edge (15, 32): Betweenness Centrality = 0.024083977025153497
Edge (15, 33): Betweenness Centrality = 0.03473955238661121
Edge (18, 32): Betweenness Centrality = 0.024083977025153497
Edge (18, 33): Betweenness Centrality = 0.03473955238661121
Edge (19, 33): Betweenness Centrality = 0.05938233879410351
Edge (20, 32): Betweenness Centrality = 0.024083977025153497
Edge (20, 33): Betweenness Centrality = 0.03473955238661121
Edge (22, 32): Betweenness Centrality = 0.024083977025153493
Edge (22, 33): Betweenness Centrality = 0.03473955238661121
Edge (23, 25): Betweenness Centrality = 0.019776193305605066
Edge (23, 27): Betweenness Centrality = 0.010536739948504653
Edge (23, 29): Betweenness Centrality = 0.00665478312537136
Edge (23, 32): Betweenness Centrality = 0.022341057635175278
Edge (23, 33): Betweenness Centrality = 0.03266983561101209
Edge (24, 25): Betweenness Centrality = 0.0042186571598336305
Edge (24, 27): Betweenness Centrality = 0.018657159833630418
Edge (24, 31): Betweenness Centrality = 0.040106951871657755
Edge (25, 31): Betweenness Centrality = 0.04205783323430383
Edge (26, 29): Betweenness Centrality = 0.004532722179781003
Edge (26, 33): Betweenness Centrality = 0.0542908072319837
Edge (27, 33): Betweenness Centrality = 0.030477039300568713
Edge (28, 31): Betweenness Centrality = 0.0148544266191325
Edge (28, 33): Betweenness Centrality = 0.024564977506153975
Edge (29, 32): Betweenness Centrality = 0.023328523328523323
Edge (29, 33): Betweenness Centrality = 0.029807882749059215
Edge (30, 32): Betweenness Centrality = 0.01705288175876411
Edge (30, 33): Betweenness Centrality = 0.02681436210847975
Edge (31, 32): Betweenness Centrality = 0.04143394731630026
Edge (31, 33): Betweenness Centrality = 0.05339388280564752
Edge (32, 33): Betweenness Centrality = 0.008225108225108224
```

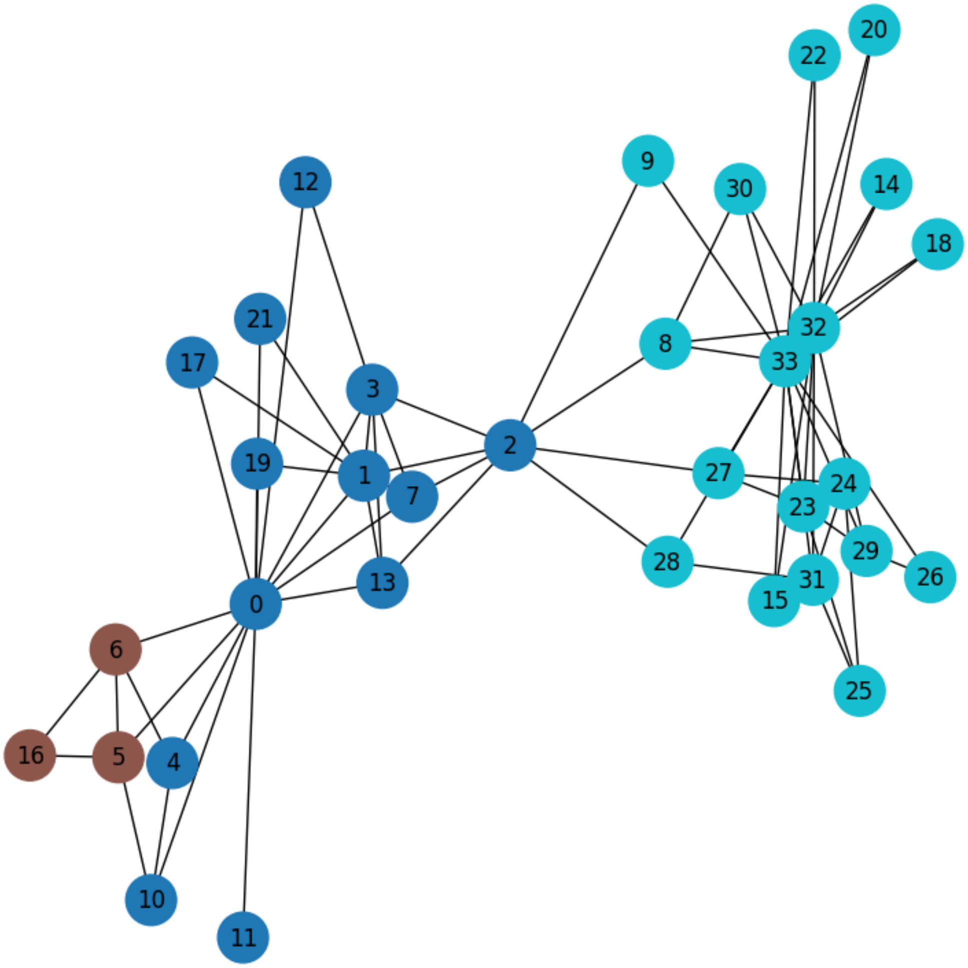
0.0.4 2.b. Remove the edge with the highest betweenness centrality.

[11]:
```
num_partitions = 3
for _ in range(num_partitions):
    remove_edge_with_high_betweenness(G)
    draw_graph_with_partition(G)
```

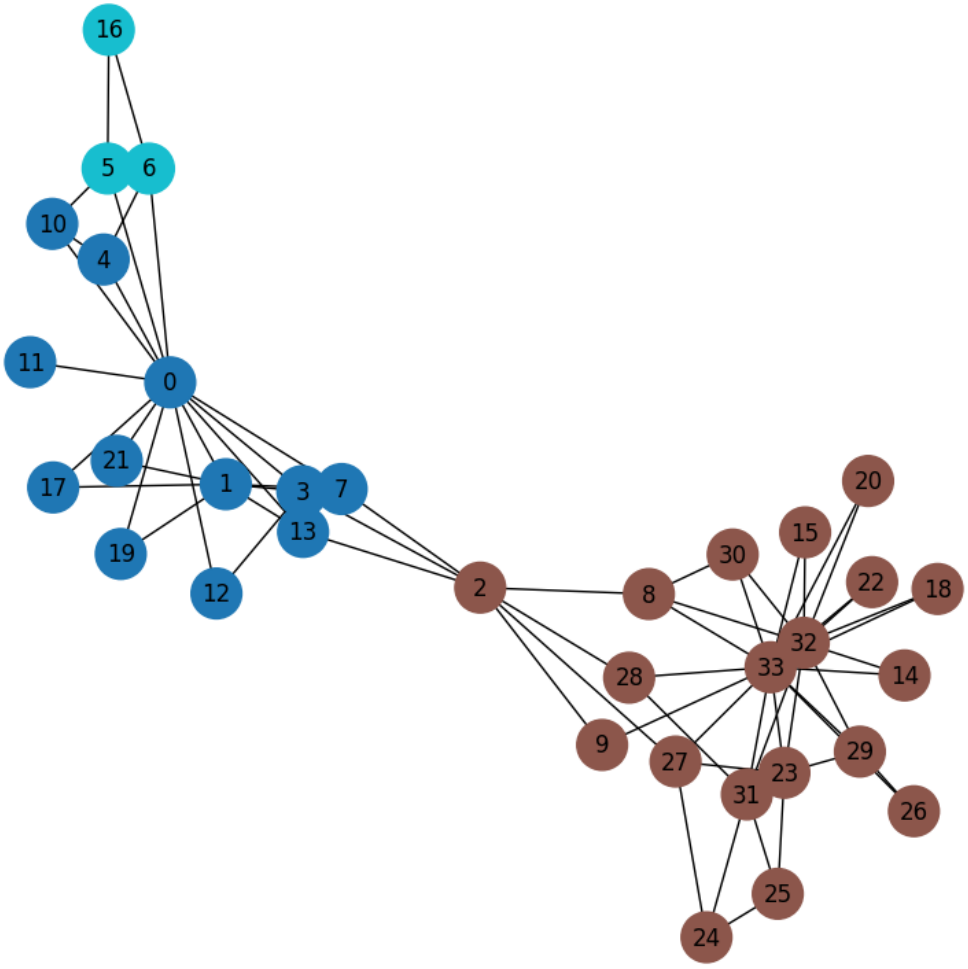
```
Removing edge (2, 32) with highest betweenness centrality
[{0, 1, 2, 3, 4, 7, 10, 11, 12, 13, 17, 19, 21}, {16, 5, 6}, {32, 33, 8, 9, 14,
15, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31}]
```



```
Removing edge (1, 30) with highest betweenness centrality
[{0, 1, 2, 3, 4, 7, 10, 11, 12, 13, 17, 19, 21}, {16, 5, 6}, {32, 33, 8, 9, 14,
15, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31}]
```



```
Removing edge (1, 2) with highest betweenness centrality
[{0, 1, 3, 4, 7, 10, 11, 12, 13, 17, 19, 21}, {2, 8, 9, 14, 15, 18, 20, 22, 23,
24, 25, 26, 27, 28, 29, 30, 31, 32, 33}, {16, 5, 6}]
```



```
[8]: calculate_edge_betweenness(G)
```

```
Edge (0, 1): Betweenness Centrality = 0.029010695187165772
Edge (0, 3): Betweenness Centrality = 0.024257278669043366
Edge (0, 4): Betweenness Centrality = 0.0522875816993464
Edge (0, 5): Betweenness Centrality = 0.07813428401663694
Edge (0, 6): Betweenness Centrality = 0.07813428401663694
Edge (0, 7): Betweenness Centrality = 0.026387501387501382
Edge (0, 8): Betweenness Centrality = 0.13782067605597018
Edge (0, 10): Betweenness Centrality = 0.0522875816993464
Edge (0, 11): Betweenness Centrality = 0.058823529411764705
Edge (0, 12): Betweenness Centrality = 0.03715304303539596
Edge (0, 13): Betweenness Centrality = 0.07645491763138822
Edge (0, 17): Betweenness Centrality = 0.029734317969612078
Edge (0, 19): Betweenness Centrality = 0.07076567223626047
Edge (0, 21): Betweenness Centrality = 0.029734317969612085
Edge (1, 2): Betweenness Centrality = 0.04866750896162661
Edge (1, 3): Betweenness Centrality = 0.0077243018419489
Edge (1, 7): Betweenness Centrality = 0.007477816301345713
Edge (1, 13): Betweenness Centrality = 0.014573988103399872
Edge (1, 17): Betweenness Centrality = 0.02908921144215263
Edge (1, 19): Betweenness Centrality = 0.017693417693417694
Edge (1, 21): Betweenness Centrality = 0.029089211442152624
Edge (1, 30): Betweenness Centrality = 0.03690492514021926
Edge (2, 3): Betweenness Centrality = 0.04199770817417876
Edge (2, 7): Betweenness Centrality = 0.038003336532748296
Edge (2, 8): Betweenness Centrality = 0.01749769838005132
Edge (2, 9): Betweenness Centrality = 0.026142974672386434
Edge (2, 13): Betweenness Centrality = 0.01601225571813807
Edge (2, 27): Betweenness Centrality = 0.05063416975181681
Edge (2, 28): Betweenness Centrality = 0.02993003728297846
Edge (2, 32): Betweenness Centrality = 0.05547459076870843
Edge (3, 7): Betweenness Centrality = 0.0033838710309298547
Edge (3, 12): Betweenness Centrality = 0.021670486376368717
Edge (3, 13): Betweenness Centrality = 0.017414938003173293
Edge (4, 6): Betweenness Centrality = 0.0047534165181224
Edge (4, 10): Betweenness Centrality = 0.0029708853238265
Edge (5, 6): Betweenness Centrality = 0.0029708853238265003
Edge (5, 10): Betweenness Centrality = 0.0047534165181224
Edge (5, 16): Betweenness Centrality = 0.029411764705882353
Edge (6, 16): Betweenness Centrality = 0.029411764705882353
Edge (8, 30): Betweenness Centrality = 0.00980392156862745
Edge (8, 32): Betweenness Centrality = 0.059880315762668695
Edge (8, 33): Betweenness Centrality = 0.06890805273158214
Edge (9, 33): Betweenness Centrality = 0.03431290931290931
Edge (13, 33): Betweenness Centrality = 0.10208533296768586
Edge (14, 32): Betweenness Centrality = 0.021360665478312533
Edge (14, 33): Betweenness Centrality = 0.03746286393345216
Edge (15, 32): Betweenness Centrality = 0.021360665478312533
Edge (15, 33): Betweenness Centrality = 0.03746286393345216
Edge (18, 32): Betweenness Centrality = 0.021360665478312533
Edge (18, 33): Betweenness Centrality = 0.03746286393345216
Edge (19, 33): Betweenness Centrality = 0.09142997525350469
Edge (20, 32): Betweenness Centrality = 0.021360665478312533
Edge (20, 33): Betweenness Centrality = 0.03746286393345216
Edge (22, 32): Betweenness Centrality = 0.021360665478312533
Edge (22, 33): Betweenness Centrality = 0.03746286393345216
Edge (23, 25): Betweenness Centrality = 0.029962847609906435
Edge (23, 27): Betweenness Centrality = 0.010085339497104204
Edge (23, 29): Betweenness Centrality = 0.00665478312537136
Edge (23, 32): Betweenness Centrality = 0.024438143555790607
Edge (23, 33): Betweenness Centrality = 0.044756060932531525
Edge (24, 25): Betweenness Centrality = 0.00531494649141708
Edge (24, 27): Betweenness Centrality = 0.03686574863045451
Edge (24, 31): Betweenness Centrality = 0.022994652406417113
Edge (25, 31): Betweenness Centrality = 0.02722963311198605
Edge (26, 29): Betweenness Centrality = 0.0045412104235633645
Edge (26, 33): Betweenness Centrality = 0.054282318988201336
Edge (27, 33): Betweenness Centrality = 0.0388286550051256
Edge (28, 31): Betweenness Centrality = 0.010620098855392975
Edge (28, 33): Betweenness Centrality = 0.029262404262404257
Edge (29, 32): Betweenness Centrality = 0.02061370002546473
Edge (29, 33): Betweenness Centrality = 0.03253119429590018
Edge (30, 32): Betweenness Centrality = 0.019508595979184216
Edge (30, 33): Betweenness Centrality = 0.02898278192395839
Edge (31, 32): Betweenness Centrality = 0.0325189516365987
Edge (31, 33): Betweenness Centrality = 0.056996108466696706
Edge (32, 33): Betweenness Centrality = 0.008243390596331772
```

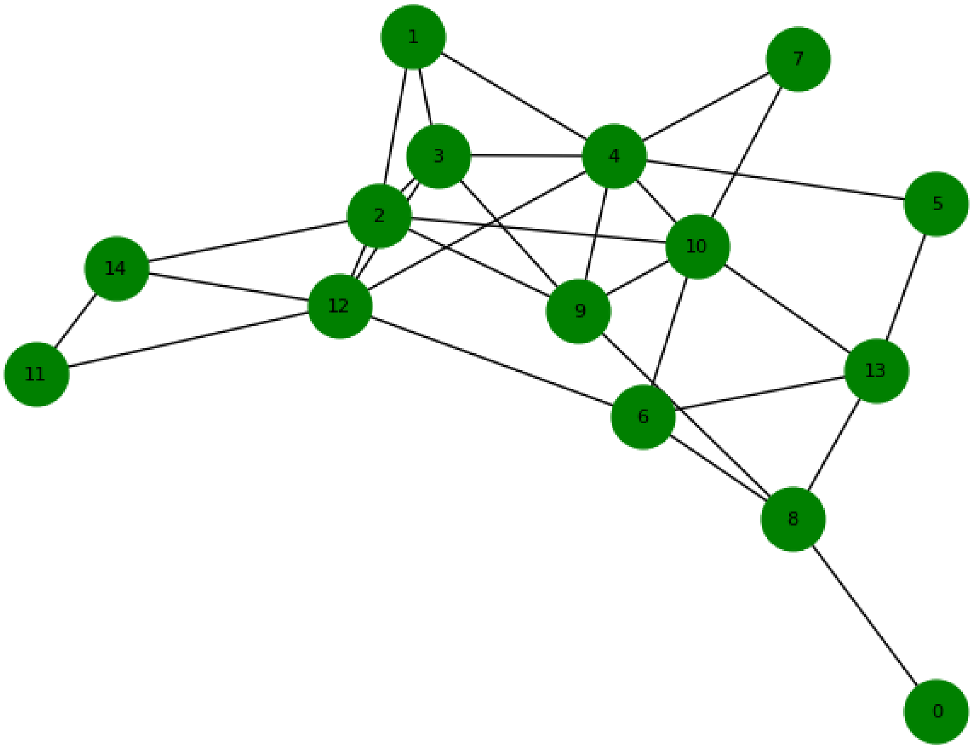
0.0.5 3. Create a random graph using GNM model.

Apply girvan-newmann algorithm to form ‘K’ partitions. N, M, and ‘K’ are input parameters.

[21]:

```
N = 15 #no of nodes
M = 30 #no of edges
random_G=nx.gnm_random_graph(N,M)

pos = nx.spring_layout(random_G)
nx.draw(random_G, pos, with_labels=True, node_size=700, node_color='Green',␣
 ↪font_size=8)
```



Output : K partitions in the form of lists

```
[48]: K = 5 # number of parition
      # Apply Girvan-Newman algorithm to form K partitions
      communities_generator = nx.community.girvan_newman(random_G)
      for _ in range(K - 1):
          partitions = next(communities_generator)
      print(list(partitions))
```

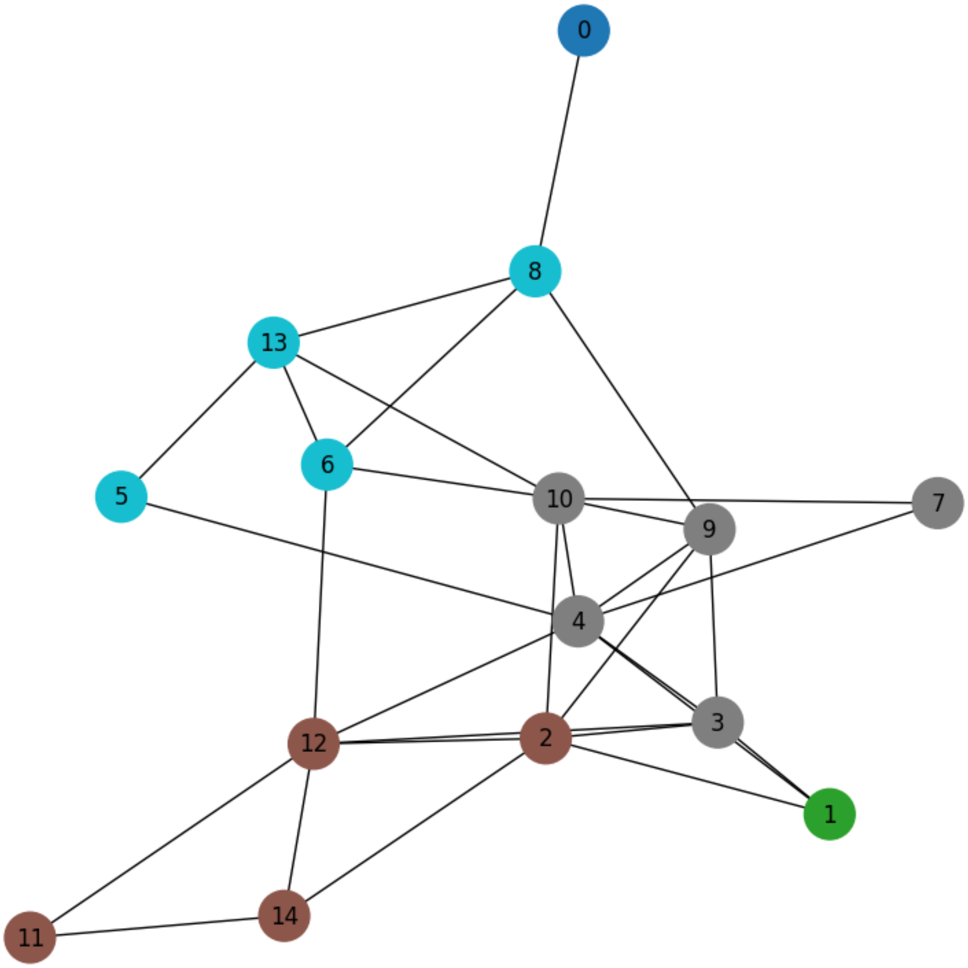
```
[{0}, {1}, {2, 11, 12, 14}, {3, 4, 7, 9, 10}, {8, 13, 5, 6}]
```

Color each of the partitions' nodes with different colors and visualize

```
[49]: node_colors = {}
      for i, partition in enumerate(partitions):
          for node in partition:
              node_colors[node] = i

      # Visualize the partitioned graph
      pos = nx.spring_layout(random_G)
      plt.figure(figsize=(8, 8))
```

```
nx.draw(random_G, pos, with_labels=True, node_size=700,␣
↪node_color=[node_colors[node] for node in random_G.nodes()], cmap=plt.cm.
↪tab10)
plt.show()
```



[ ]: