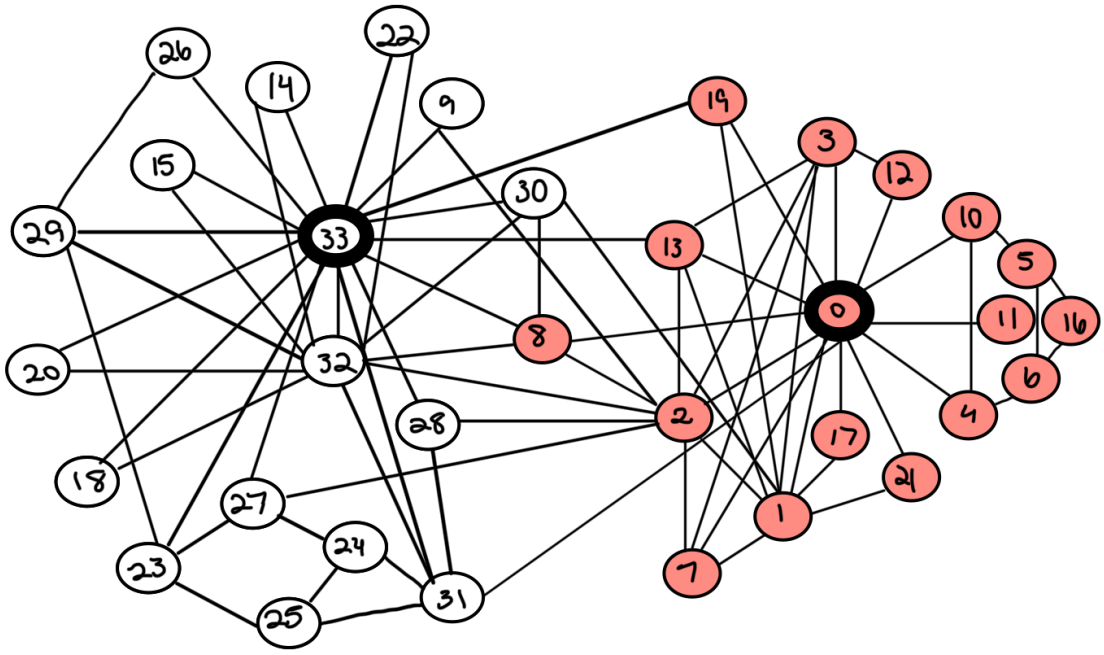

Figure 1: Hand Drawn Karate Club Graph (Before Split)

HW#5 - Graph Partitioning

Ethan Landers

Due by Sunday, November 3rd, by 11:59 PM

Q1

Q: How many nodes eventually go with John and how many with Mr. Hi?

See Figure 1; it's the original Karate Club graph before the split hand-drawn by me. Seventeen nodes eventually go with John, and seventeen go with Mr. Hi.

Q2

To generate graphs for this assignment, I used NetworkX, where each node represents a club member, and edges represent connections or relationships between members.

Also, I created a function get_node_colors() to assign colors to the nodes based on their actual faction affiliations (”Mr. Hi” or ”John”) with different colors to distinguish the two factions. See below:

```
def get_node_colors(graph):
    node_colors = []
    for node in graph.nodes(data=True):
        if node[1]['club'] == 'Mr. Hi':
            node_colors.append('salmon')
        else:
            node_colors.append('white')
    return node_colors
```

I then generated an initial graph (Figure 2) by taking a snapshot of the full, connected network before removing edges. The function used is called save_graph_snapshot(), which

- Takes a graph and color information,
- Sets up the spring layout
- Draws nodes with the specified colors and labels
- Saves the image with the given title, which in this case it is ”Initial Karate Club Graph.

The Girvan-Newman algorithm is iterative, repeatedly removing edges with the highest betweenness centrality until the graph is split into disconnected components. For each iteration, an image is generated to visualize the split. The last generated image visualizes the split into disconnected components. See Figure 3.

The below code runs the Girvan-Newman algorithm:

```
while nx.number_connected_components(working_graph) == 1:
    edge_betweenness = nx.edge_betweenness_centrality(working_graph)
    edge_to_remove = max(edge_betweenness, key=edge_betweenness.get)
    working_graph.remove_edge(*edge_to_remove)

    if nx.number_connected_components(working_graph) > 1:
        break

    save_graph_snapshot(...)

    iteration += 1
```

Q: How many iterations did it take to split the graph?

It took 11 iterations to split the graph with the Girvan-Newman algorithm.

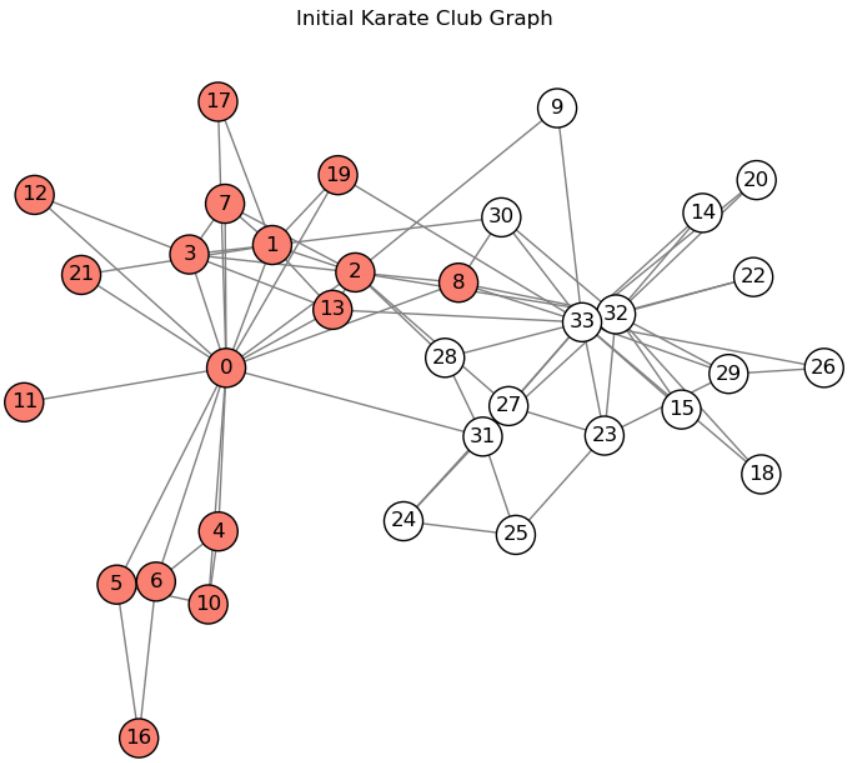

Figure 2: Initial Karate Club Graph)

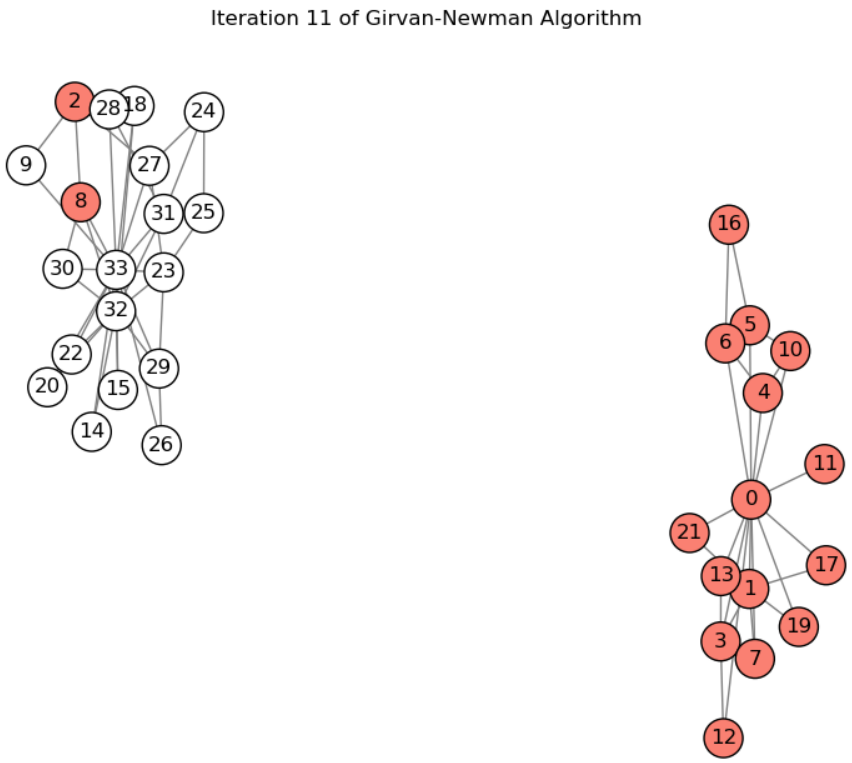

Figure 3: Girvan Newman Iteration 11

Q3

Below are the results of the comparison between the connected components of the Girvan-Newman split graph (Q2) with the connected components of the actual split Karate club graph (Q1):

```
Actual split factions:
Mr. Hi's faction: {0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 16, 17, 19, 21}
John's faction: {32, 33, 9, 14, 15, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31}

Comparison of Girvan-Newman results with actual split:
Component 1: {0, 1, 3, 4, 5, 6, 7, 10, 11, 12, 13, 16, 17, 19, 21}
Matches actual factions? No, not matched
Component 2: {2, 8, 9, 14, 15, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33}
Matches actual factions? No, not matched
```

I extracted the actual factions based on each node's attribute 'club', creating sets for the "Mr. Hi" and "John" factions.

The program's code compares the Girvan-Newman split components with the actual components to determine whether they match, then outputs "matched" or "not matched" accordingly.

Q: Did all of the same colored nodes end up in the same group? If not, what is different?

No, not all nodes from the same faction ended up in the same group. Two of Mr. Hi's faction nodes were grouped with John's in the final split (Refer to Figure 3).

References

- Centrality - NetworkX, https://networkx.org/documentation/stable/reference/algorithms/centrality.html
- Components - NetworkX, https://networkx.org/documentation/stable/reference/algorithms/component.html
- CS 432/532 - NetworkX Example, https://github.com/odu-cs432-websci/public/blob/main/432_NetworkX_example.ipynb
- Drawing - NetworkX, https://networkx.org/documentation/stable/reference/drawing.html#module-networkx.drawing.layout
- Karate Club - NetworkX, https://networkx.org/documentation/stable/auto_examples/graph/plot_karate_club.html
- Tutorial - NetworkX, https://networkx.org/documentation/stable/tutorial.html