Srividya Majeti

Assignment 5

CS 532: Introduction to Web Science
Dr. Michael Nelson
Spring 2016

March 3, 2016

Contents

1 **Question 1** 1

2 **Question 2** 21

1

Question 1

We know the result of the Karate Club (Zachary, 1977) split. Prove or disprove that the result of split could have been predicted by the weighted graph of social interactions. How well does the mathematical model represent reality? Generously document your answer with all supporting equations, code, graphs, arguments, etc. Useful sources include:

*** Original paper**

`http://aris.ss.uci.edu/~lin/76.pdf`

*** Slides**

`http://www-personal.umich.edu/~ladamic/courses/networks/si614w06/ppt/lecture18.ppt`

`http://clair.si.umich.edu/si767/papers/Week03/Community/CommunityDetection.pptx`

*** Code and data**

`http://networkx.github.io/documentation/latest/examples/graph/karate_club.html`

`http://nbviewer.ipython.org/url/courses.cit.cornell.edu/info6010/resources/11notes.ipynb`

`http://stackoverflow.com/questions/9471906/what-are-the-differences-between-community-detection-algorithms-in-igraph/9478989#9478989`

`http://stackoverflow.com/questions/5822265/are-there-implementations-of-algorithms-for-community-detection-in-graphs`

`http://konect.uni-koblenz.de/networks/ucidata-zachary`

`http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/ucidata.htm#zachary`

`https://snap.stanford.edu/snappy/doc/reference/CommunityGirvanNewman.html`

`http://igraph.org/python/doc/igraph-pysrc.html#Graph.community_edge_betweenness`

Following are the steps I have taken to solve the given problem:

- To understand about the Karate Club split (Zachary, 1977) I read the original paper from `http://aris.ss.uci.edu/~lin/76.pdf` which explains the reason behind the split and also illustrates the graphical and matrix representation of the social relationships among the 34 individuals in the karate club.
- Looking at week 6 PowerPoint slides 'Social Networks' I understood the difference between strong and weak ties, results of removing strong and weak ties and also some general approaches like 'Divisive method' and 'Agglomerative method' for partitioning a graph.
- After researching all the sources provided, I decided to solve this problem using the Divisive method proposed by 'Girvan' and 'Newman' which uses the concept of 'edge betweenness' to identify which edge to be removed.
- I downloaded the data for Karate Club in GraphML format.
- To implement the Girvan Newman algorithm and produce the graphs I installed an open source library 'python-igraph' which is a network analysis tool that has large set of graph generators, layout methods to visualize a graph, and also built-in routines to calculate centrality properties like edge and vertex betweenness.

- I wrote a Python code that reads the GraphML file and plots the Original Graph without any split which is illustrated in Figure 1.1 and a graph which distinguishes the Karate Club split based on faction which is in Figure 1.2.

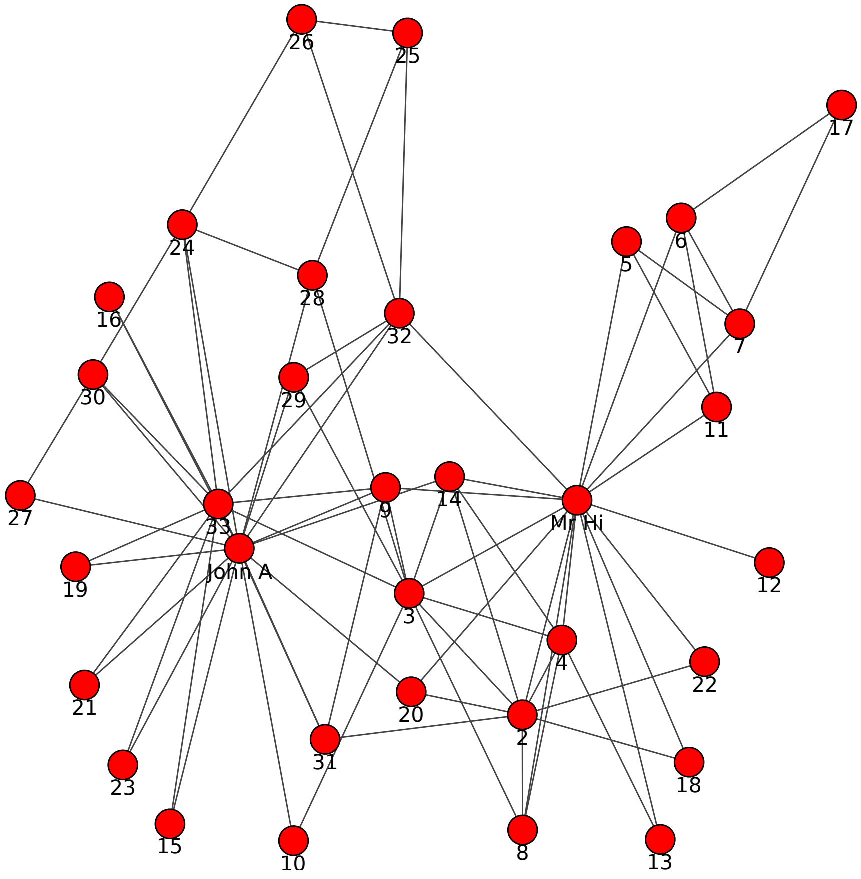

Fig. 1.1. Original Graph without any split

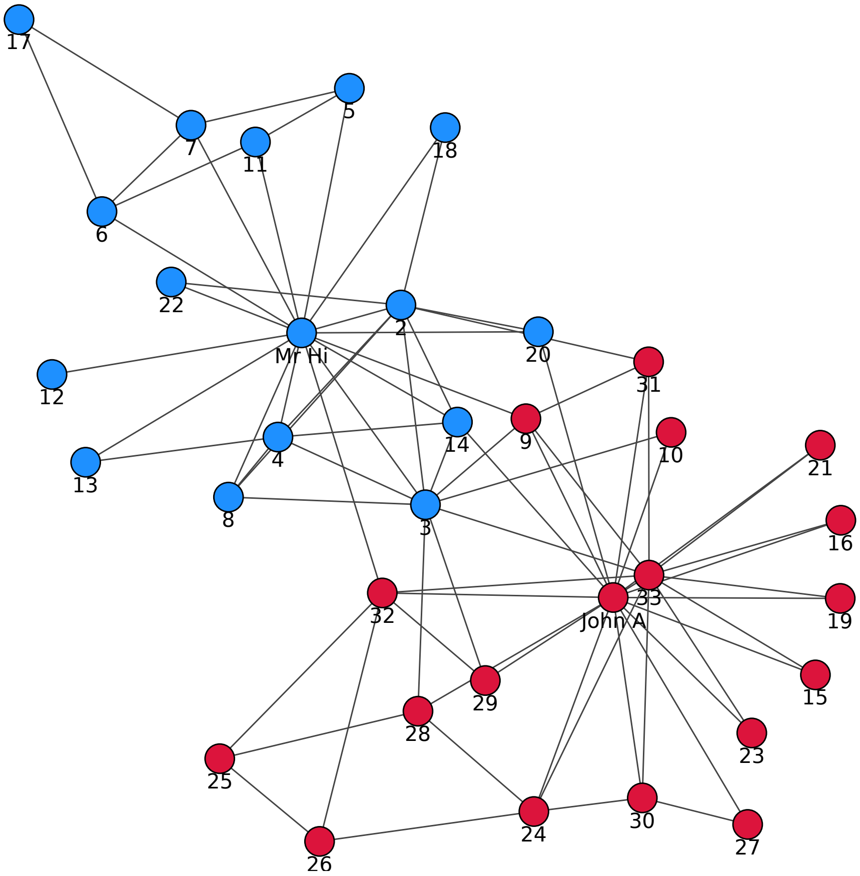

Fig. 1.2. Graph which distinguishes two groups based on Faction

- Furthermore, I calculated the edge betweenness for all the edges in the graph

- In each iteration the edges with highest betweenness is highlighted in ‘red’ and then deleted until graph is partitioned into as many regions as desired. These graphs are illustrated in Figures 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12, 1.13.

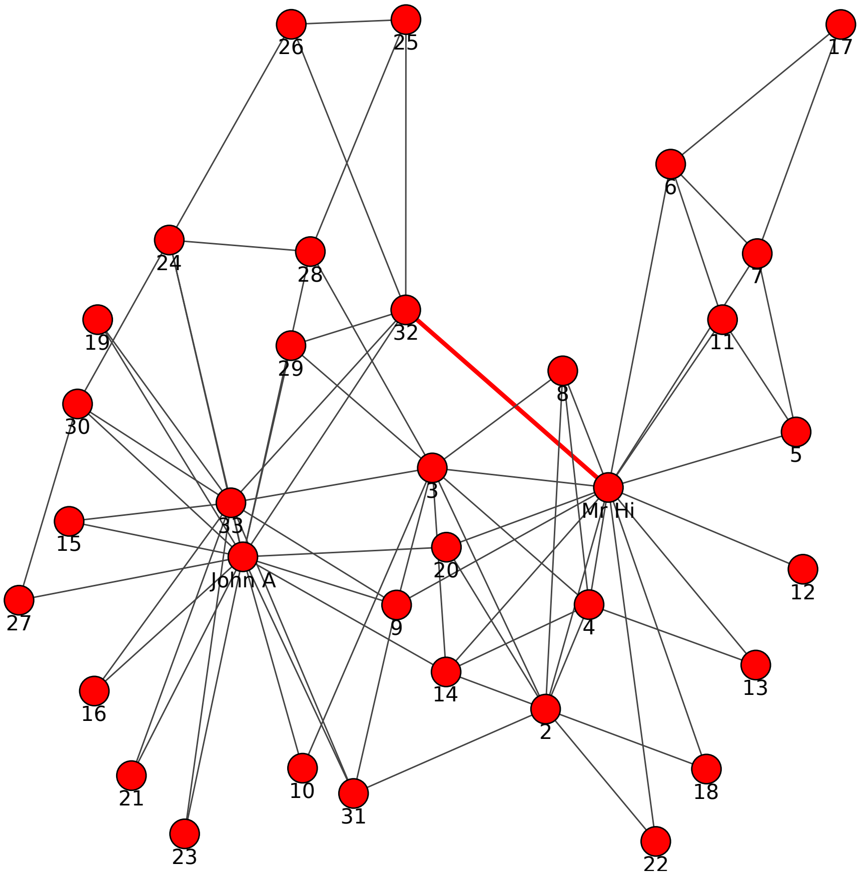

Fig. 1.3. Iteration 1

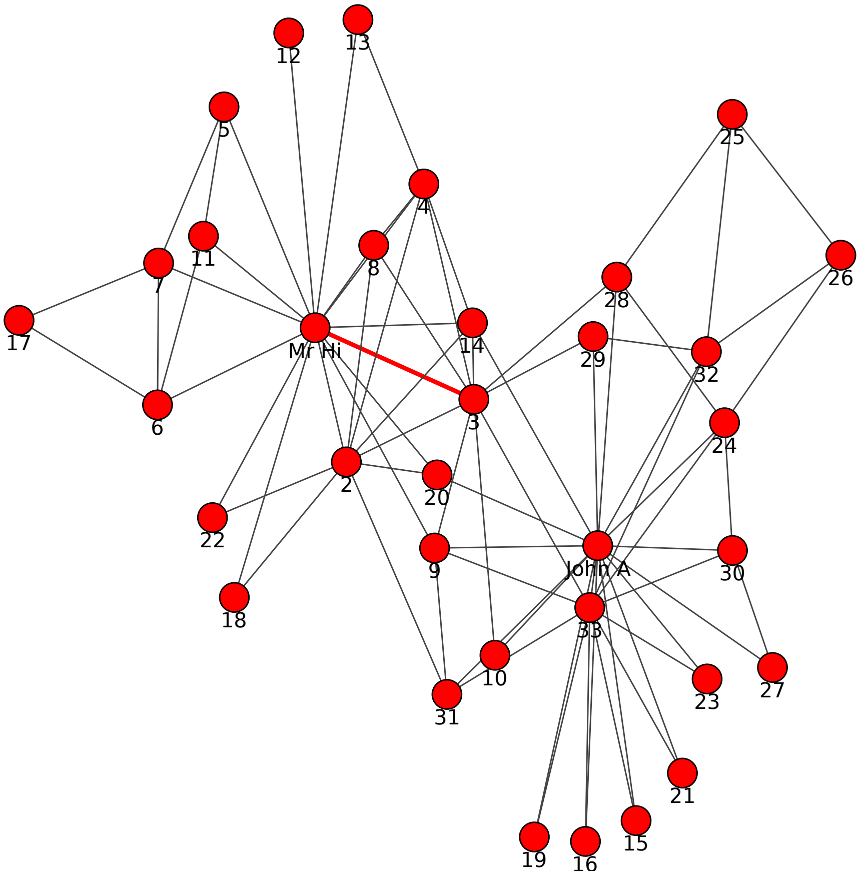

Fig. 1.4. Iteration 2

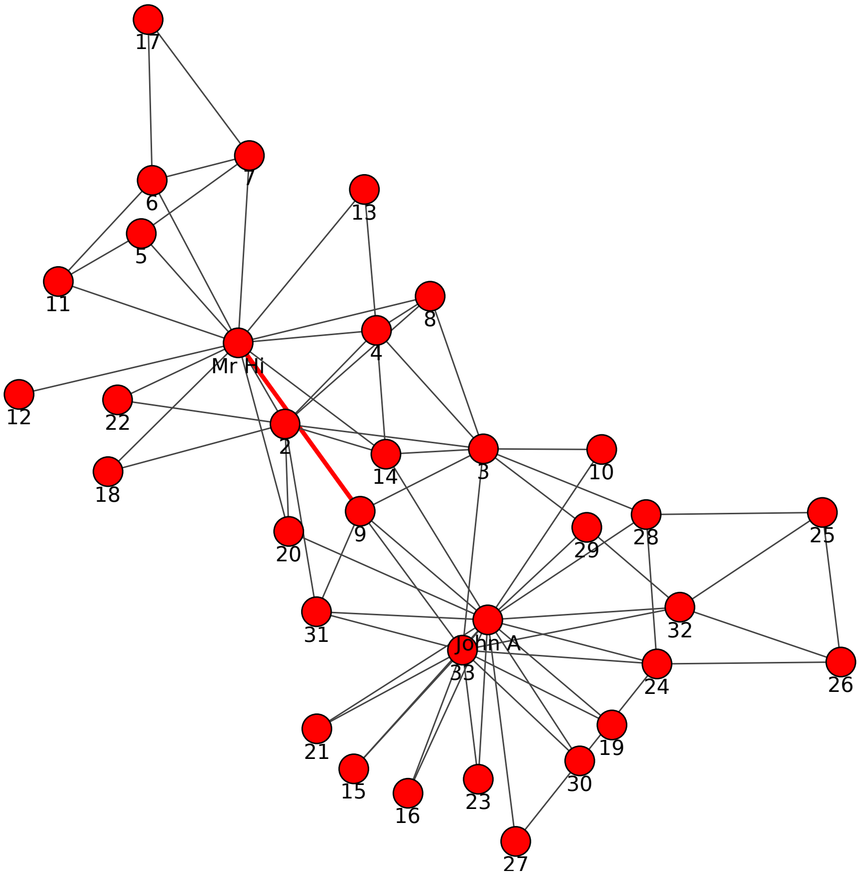

Fig. 1.5. Iteration 3

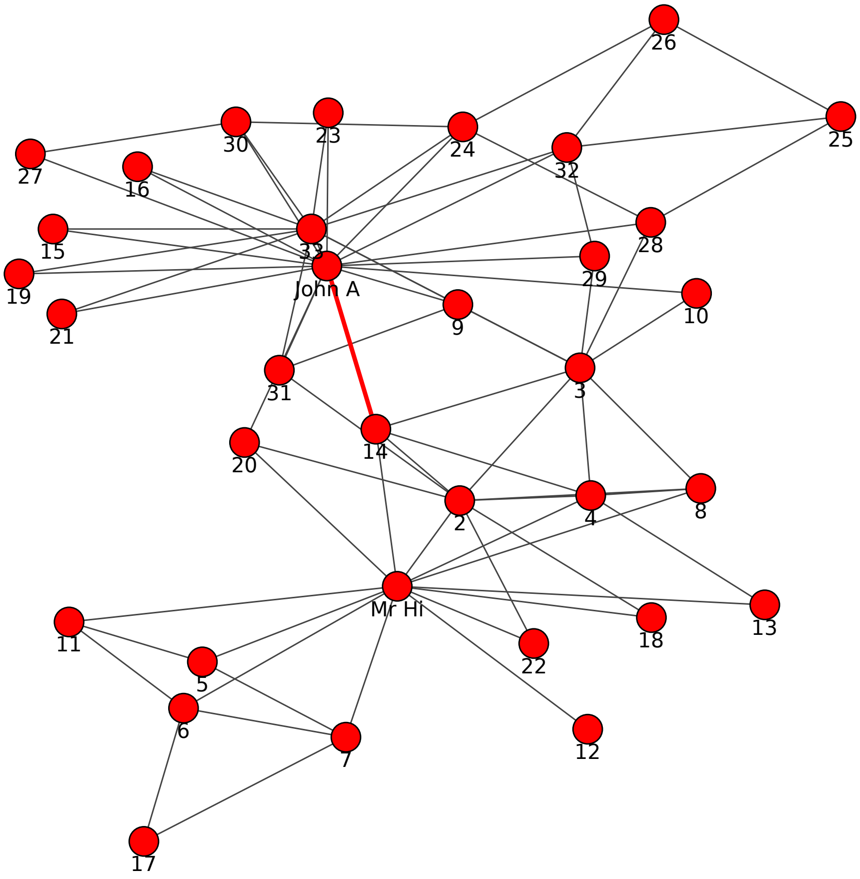

Fig. 1.6. Iteration 4

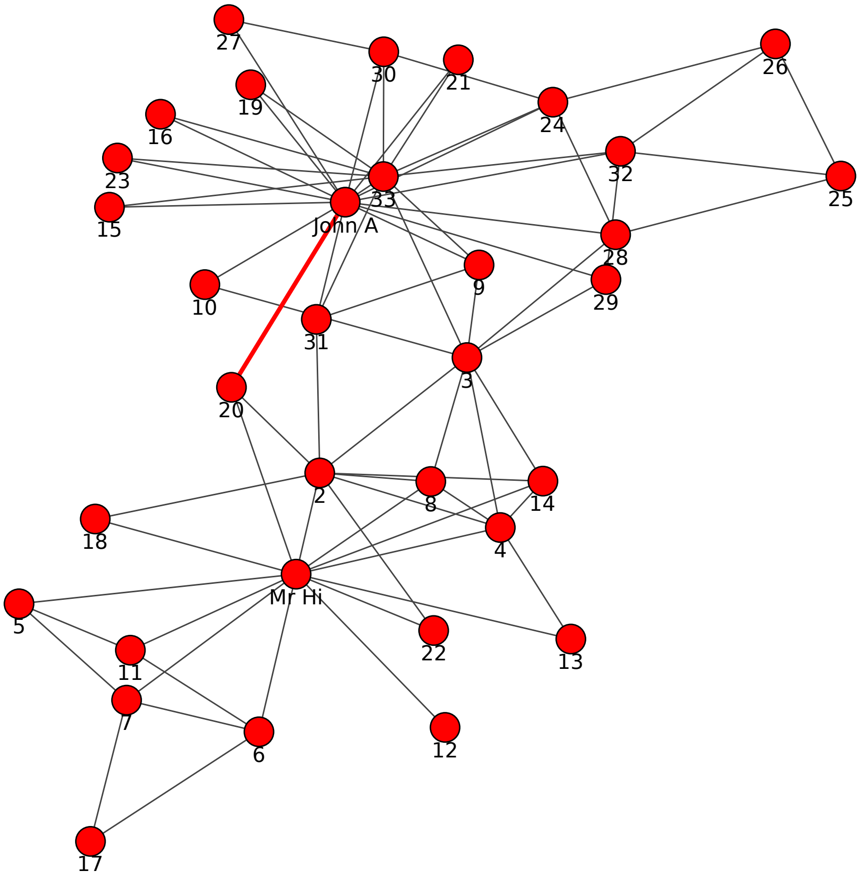

Fig. 1.7. Iteration 5

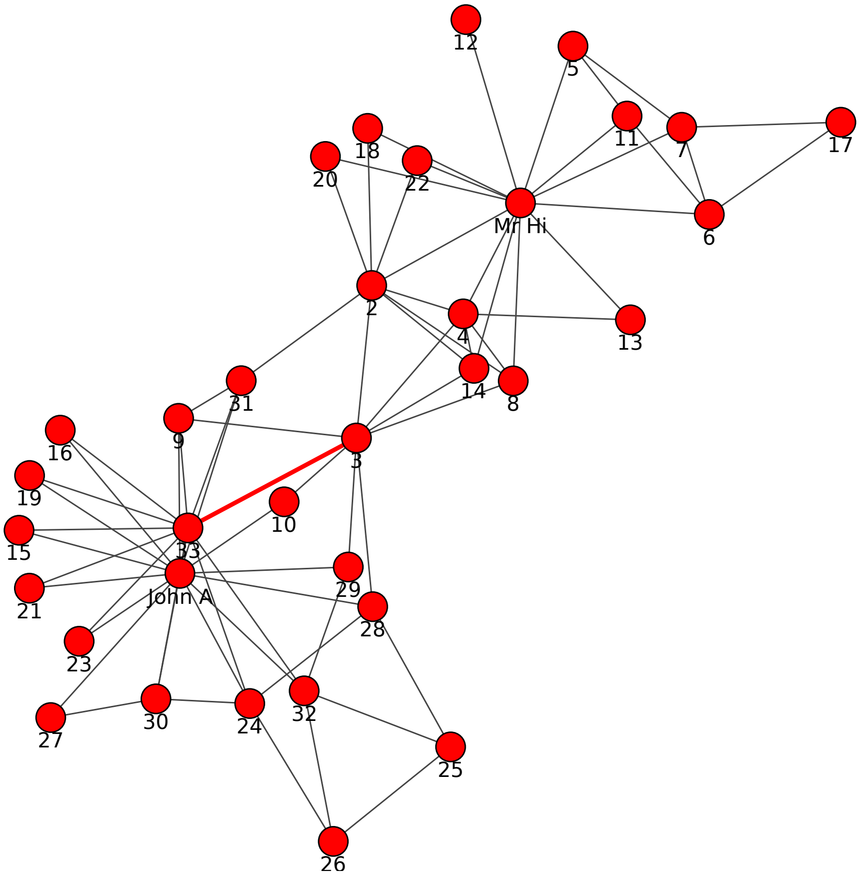

Fig. 1.8. Iteration 6

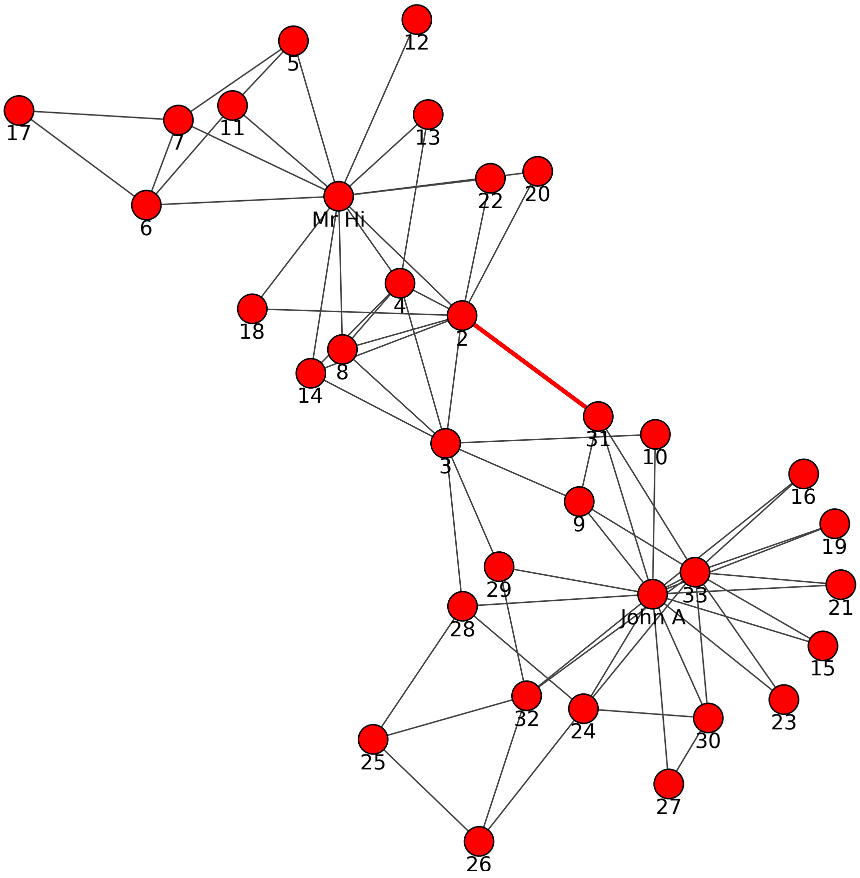

Fig. 1.9. Iteration 7

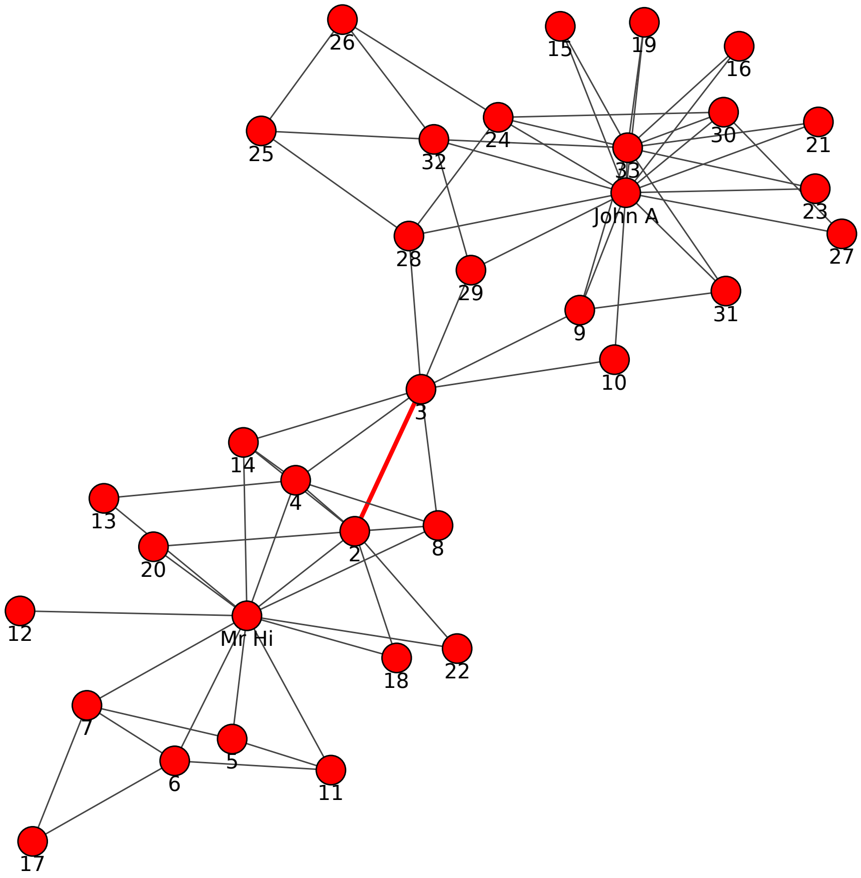

Fig. 1.10. Iteration 8

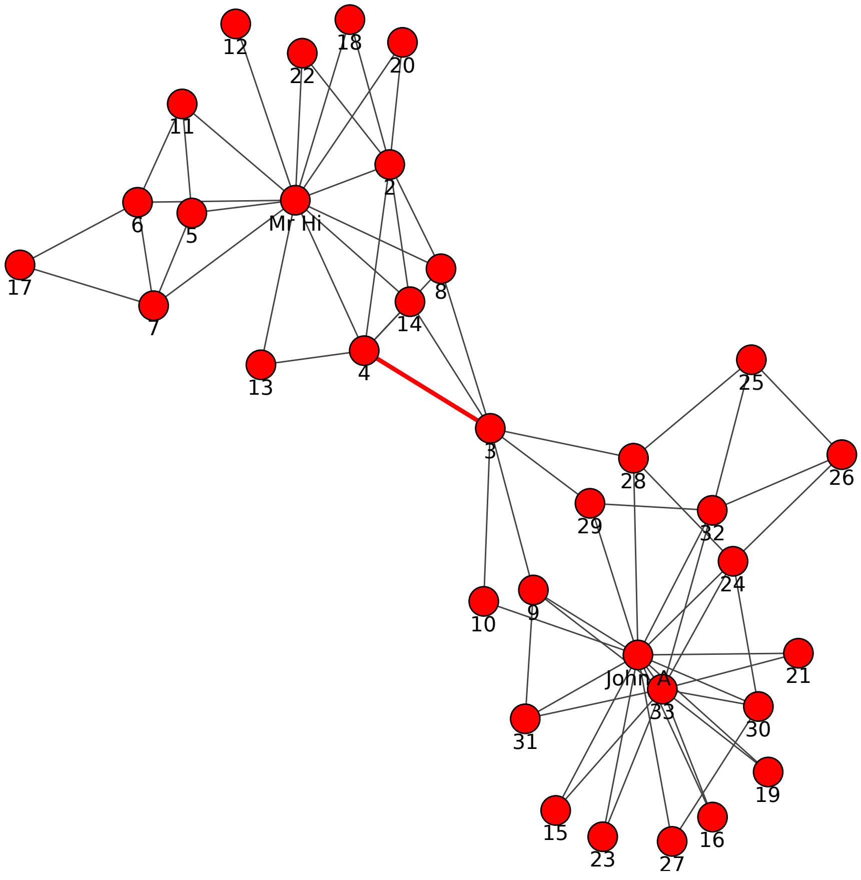

Fig. 1.11. Iteration 9

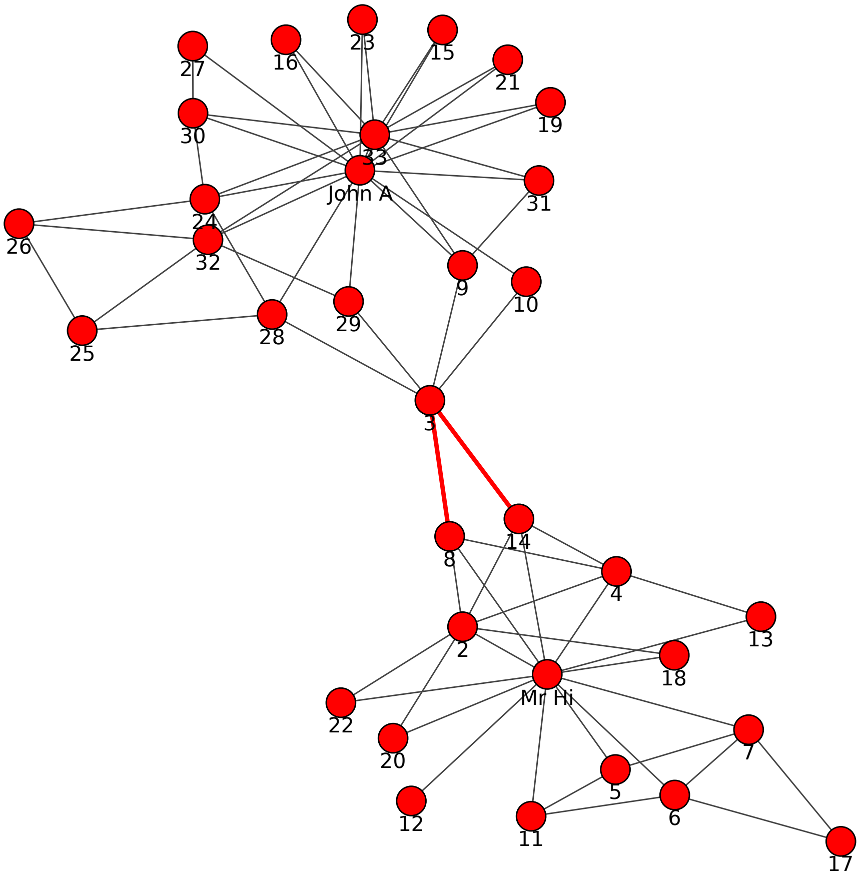

Fig. 1.12. Iteration 10

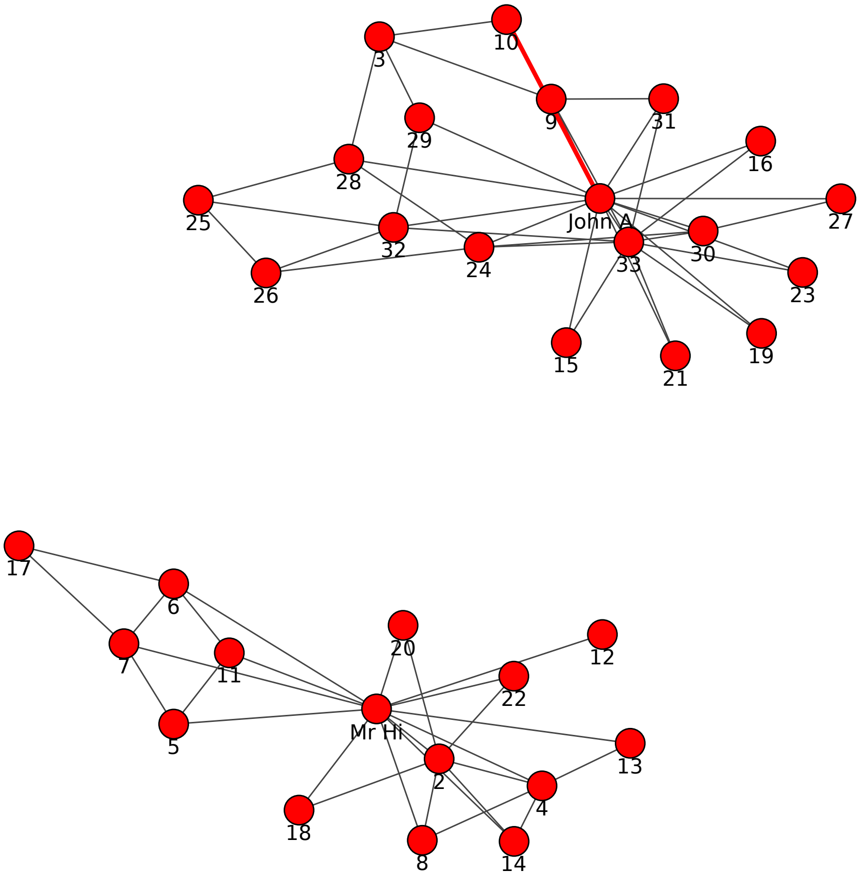

Fig. 1.13. Iteration 11

- After 11 iterations, the graph is partitioned into 2 groups which is illustrated in Figure 1.14.

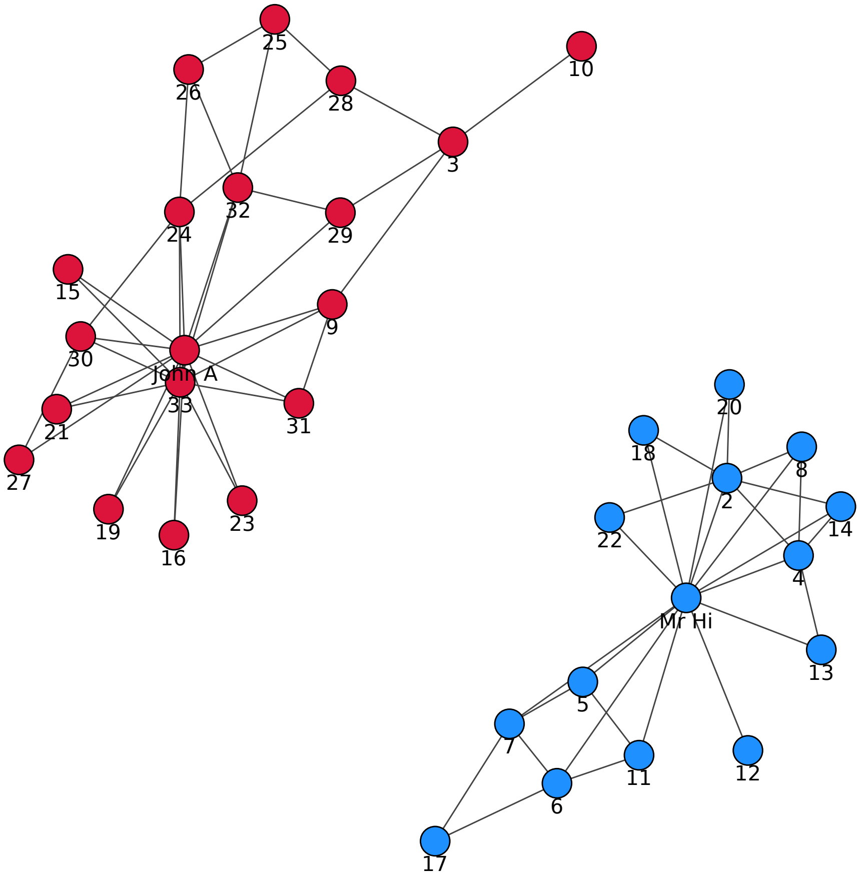

Fig. 1.14. Output of Girvan Newman algorithm which divides the Karate Club Graph divided into 2 groups

- Comparison between the 'Faction graph' and the resultant Karate Club split graph generated from 'Girvan Newman' algorithm is illustrated in 1

Faction Graph	Girvan Newman Graph	Faction Graph	Girvan Newman Graph
Group 1		Group 2	
John A	John A	Mr.Hi	Mr.Hi
-	3	2	2
9	9	3	-
10	10	4	4
15	15	5	5
16	16	6	6
19	19	7	7
21	21	8	8
23	23	11	11
24	24	12	12
25	25	13	13
26	26	14	14
27	27	17	17
28	28	18	18
29	29	20	20
30	30	22	22
31	31		
32	32		
33	33		

- By looking at the table, We can say that all the nodes are same except from node 3. By this we can conclude that the mathematical model matches almost with the reality.

- This code is listed in Listing 1.1 **Code Listing**

```
from igraph import *
from xml.etree import ElementTree
import sys

karateClubGraph = Graph.Read_GraphML("karate.GraphML")
folderName = "graphs"
color_dict={1:"#1E90FF" , 2:"#DC143C", 3:"black", 4:"green" , 5:"lightblue"}

def savePlot(filename, visual_style):
        plot(karateClubGraph, './'+folderName+ '/'+filename + '.pdf', **visual_style)

def plotGraph(filename, vertColorChanged=False):
        visual_style = {}
        visual_style["vertex_size"] = 20
        visual_style["vertex_label"] = karateClubGraph.vs["name"]
        visual_style["vertex_label_dist"] = 1
        if vertColorChanged == True:
                print "Color Changed"
                visual_style["vertex_color"] = [vert["vertex_color"] for vert in karateClubGraph.vs]
        visual_style['edge_color'] = [edgeColor for edgeColor in karateClubGraph.es['edge_color']]
        visual_style['edge_width'] = [edgeWidth for edgeWidth in karateClubGraph.es['edge_width']]
        if vertColorChanged == True:
                print visual_style
        layout = karateClubGraph.layout("rt_circular")
        savePlot(filename, visual_style)

def plotFactionKarateClubGraph():
        visual_style = {}
        visual_style["vertex_size"] = 20
        visual_style["vertex_label"] = karateClubGraph.vs["name"]
        visual_style["vertex_label_dist"] = 1
        visual_style["vertex_color"] = [color_dict[node] for node in karateClubGraph.vs["Faction"]]
        layout = karateClubGraph.layout("rt_circular")
        savePlot('FactionGraph', visual_style)

def findEdgeBetweenness():
        numberOfclusters = 0
        count = 0
```

```python
    while True:
        count += 1
        edgeBetweennessValue =karateClubGraph.edge_betweenness()
        max_edgeBetweennessValue = max(edgeBetweennessValue)
        edge = [idx for idx, edgeBetweenness in enumerate(edgeBetweennessValue) if edgeBetweenness == max_edgeBetweennessValue]
        for idx in edge:
            karateClubGraph.es[idx]["edge_color"] = "red"
            karateClubGraph.es[idx]["edge_width"] = 3

        plotGraph ("EdgeHighlightedGraph"+str(count))
        karateClubGraph.delete_edges(edge)
        clusters = karateClubGraph.clusters('weak')
        numberOfclusters = len(clusters)
        print clusters
        print numberOfclusters
        print "count:",count
        c=1
        for cluster in clusters:
            for vertex in cluster:
                karateClubGraph.vs[vertex]["vertex_color"] = color_dict[c]
            c = c+1
        plotGraph ("Graph with"+str(numberOfclusters)+"Groups", True)
        clusterCount = 1
        if numberOfclusters > 4:
            break

plotGraph("OriginalGraph")
plotFactionKarateClubGraph()
findEdgeBetweenness()
```

Listing 1.1. Python code which implements Girvan Newman algorithm to divide Karate Club Graph into groups of ‘2’ ‘3’ ‘4’ and ‘5’

2

Question 2

We know the group split in two different groups. Suppose the disagreements in the group were more nuanced – what would the clubs look like if they split into groups of 3, 4, and 5?

- I continued iterating till I get groups of 3, 4 and 5.

- I got the partitioned graph with 3 groups in 14th iteration. The resultant graph for each iteration with highlighted edges is illustrated in Figures 2.1, 2.2, 2.3 and the output graph with 3 groups is in Figure 2.4.

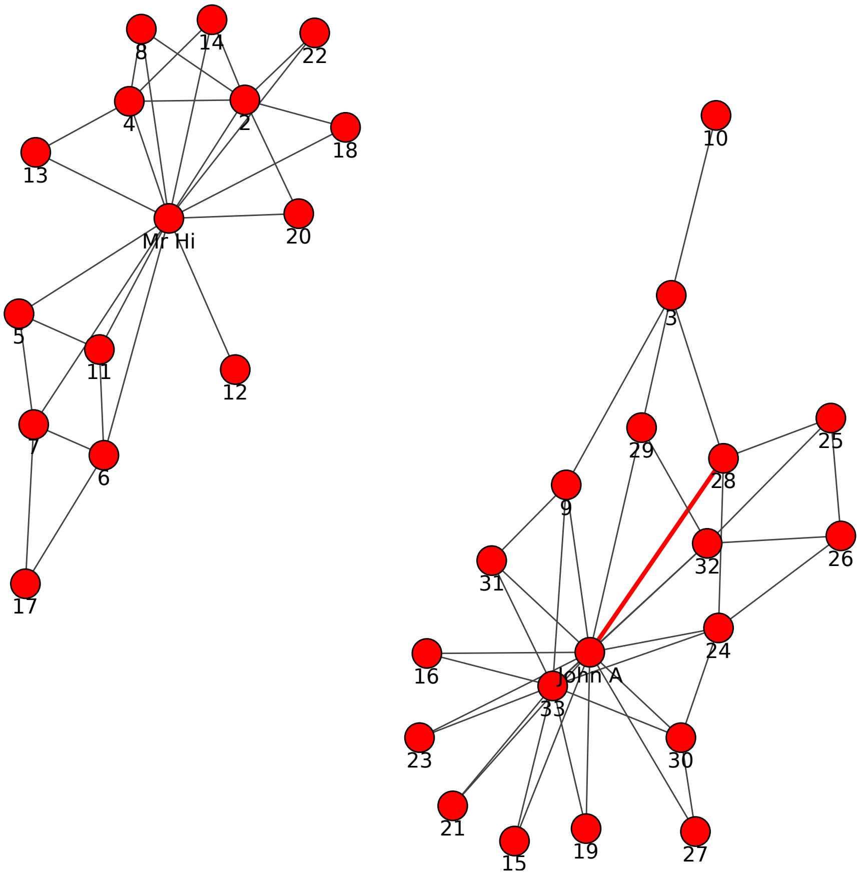

Fig. 2.1. Iteration 12

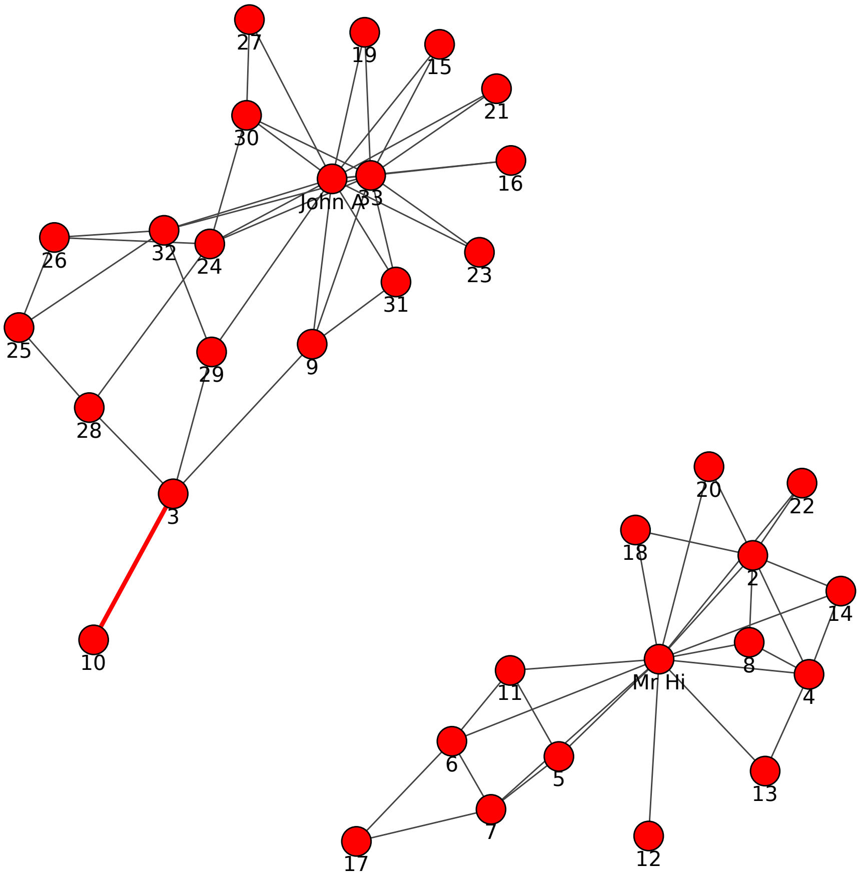

Fig. 2.2. Iteration 13

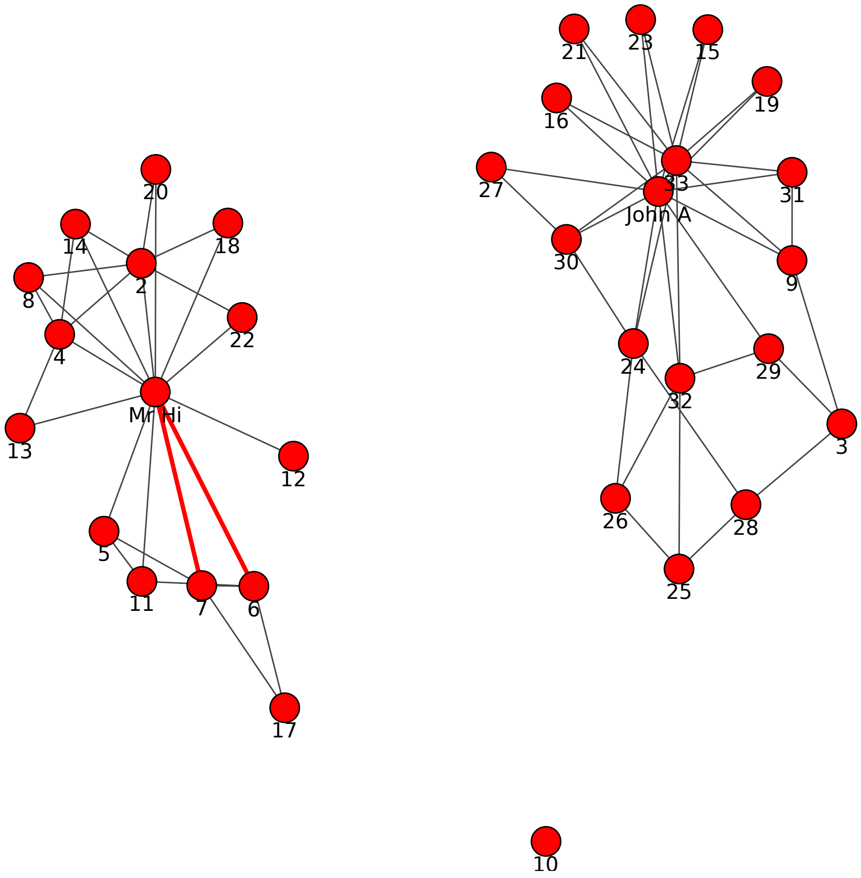

Fig. 2.3. Iteration 14

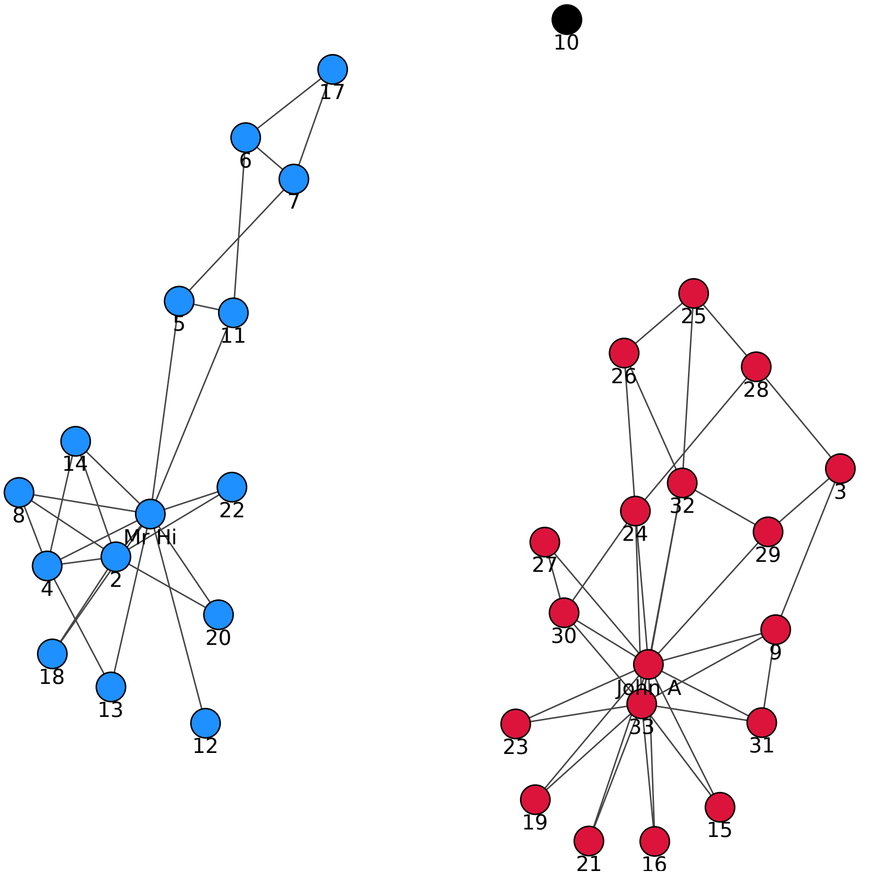

Fig. 2.4. Output of Girvan Newman algorithm which divides the Karate Club Graph divided into 3 groups

- I got the partitioned graph with 4 groups in 16th iteration. The resultant graph for each iteration with highlighted edges is illustrated in Figures 2.1, 2.2, 2.3 and the output graph with 3 groups is in Figure 2.4.

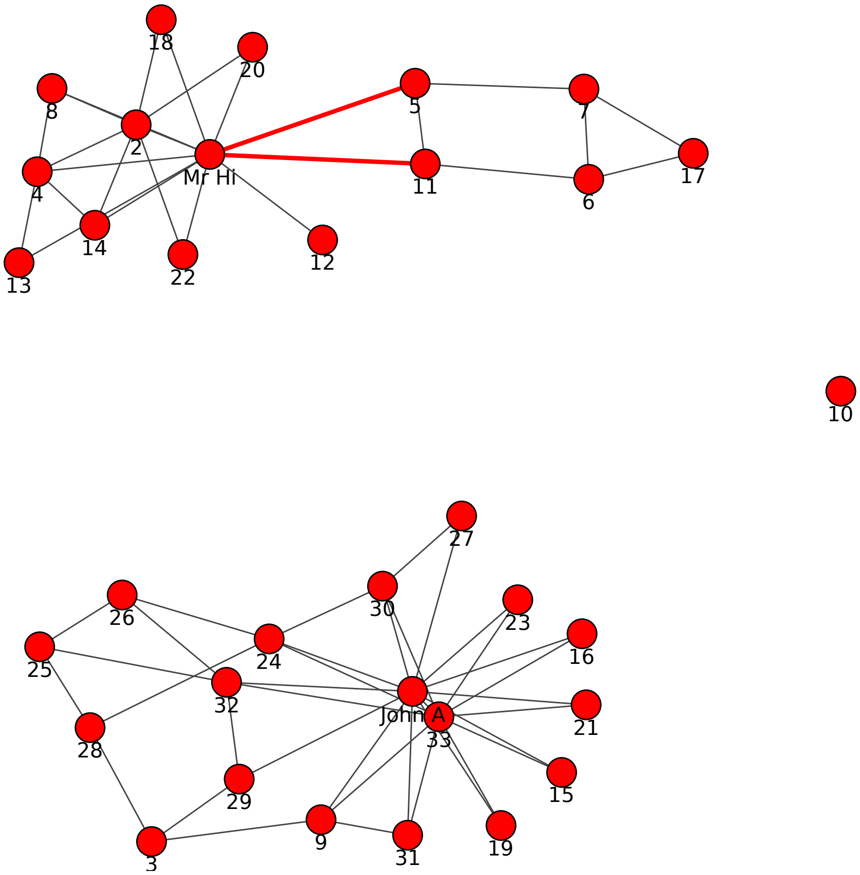

Fig. 2.5. Iteration 15

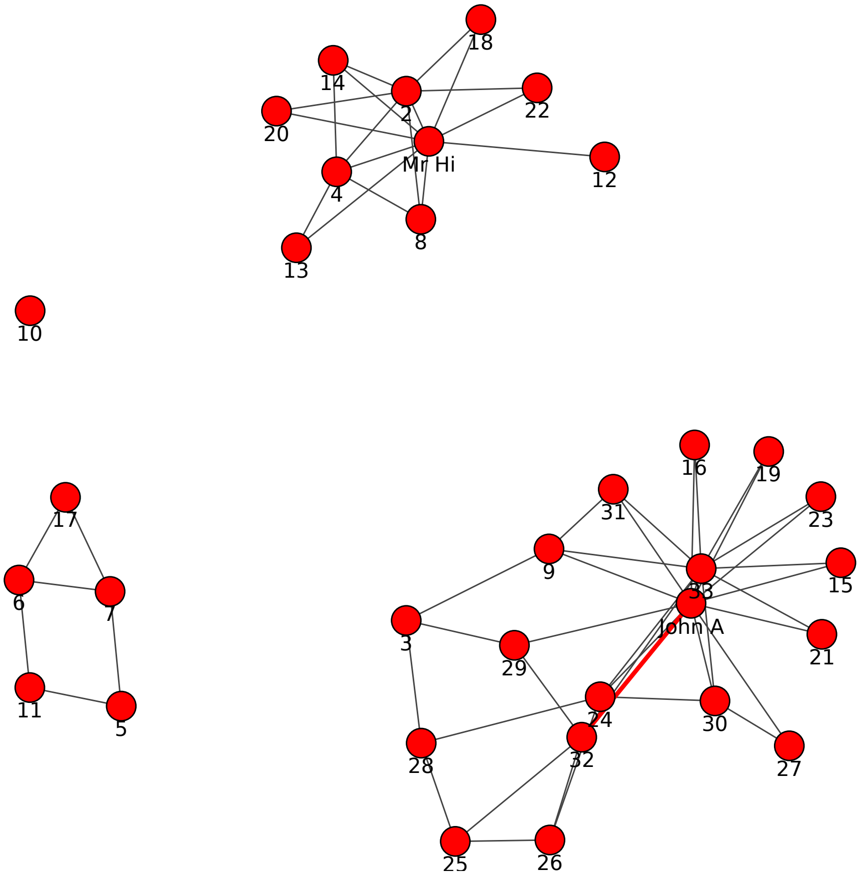

Fig. 2.6. Iteration 16

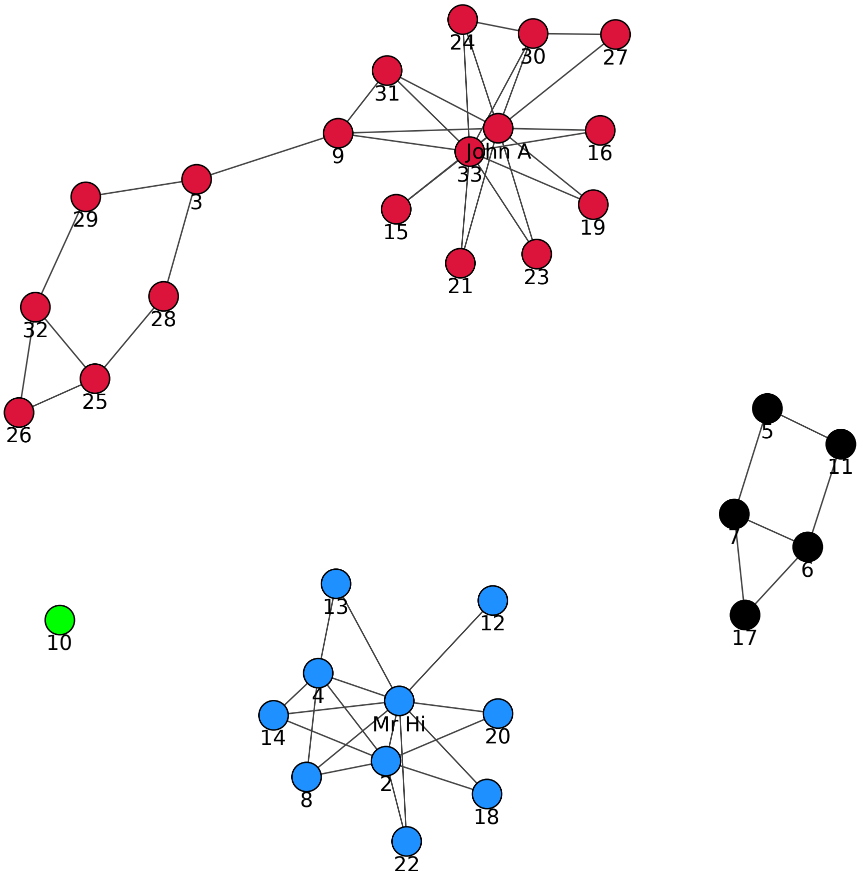

Fig. 2.7. Output of Girvan Newman algorithm which divides the Karate Club Graph divided into 4 groups

- I got the partitioned graph with 3 groups in 21st iteration. The resultant graph for each iteration with highlighted edges is illustrated in Figures 2.1, 2.2, 2.3 and the output graph with 3 groups is in Figure 2.4.

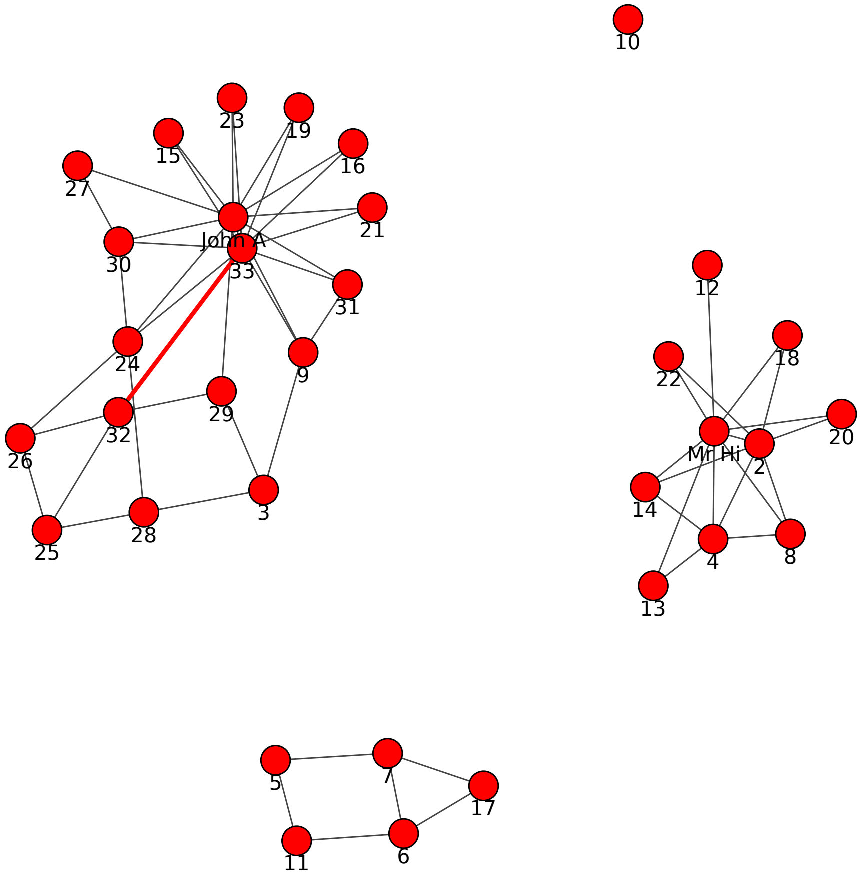

Fig. 2.8. Iteration 17

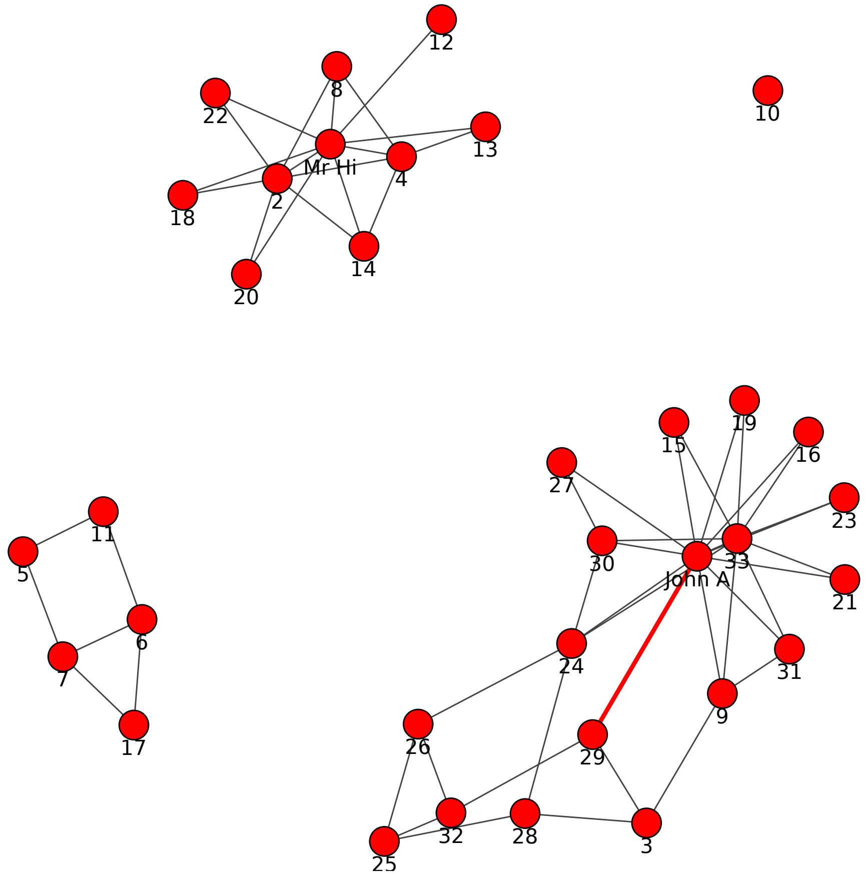

Fig. 2.9. Iteration 18

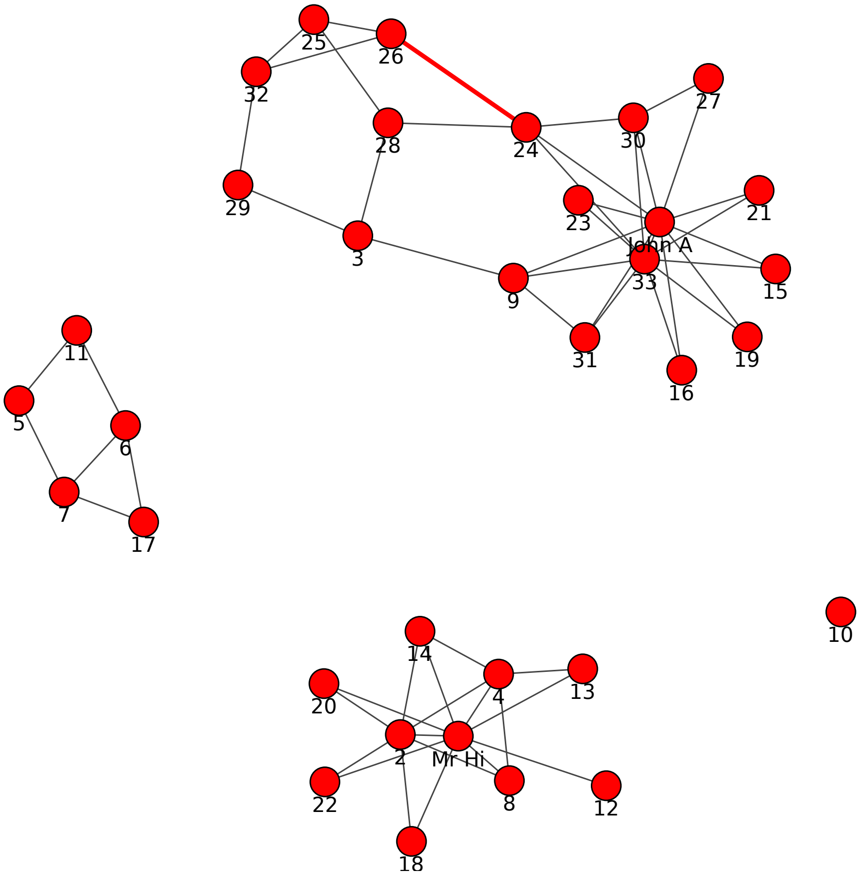

Fig. 2.10. Iteration 19

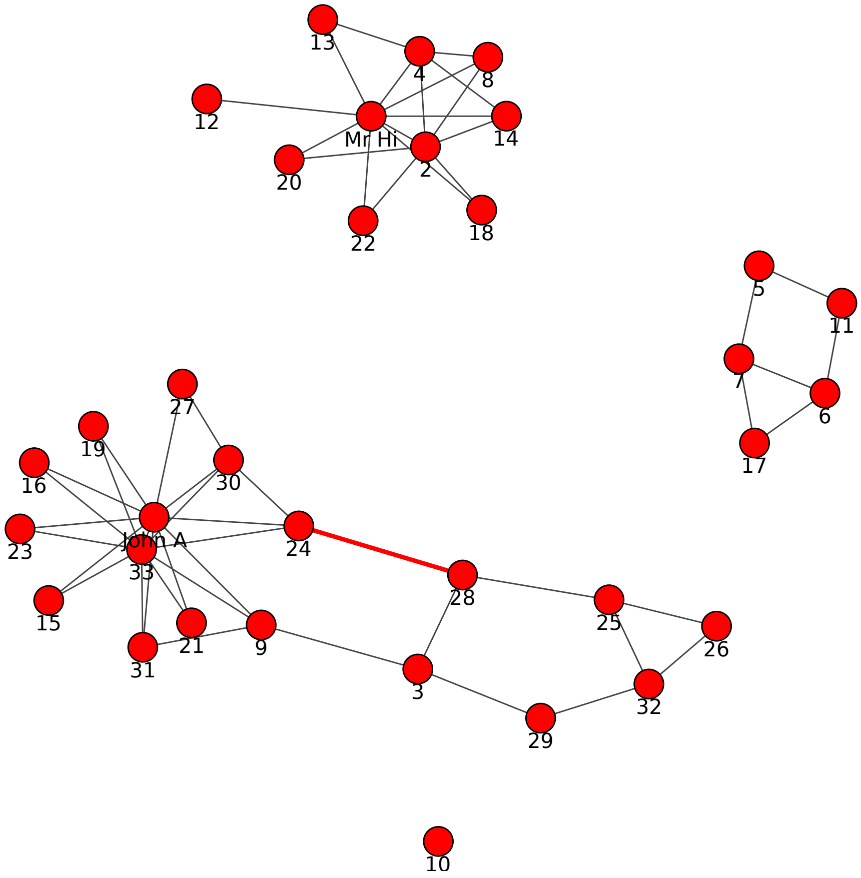

Fig. 2.11. Iteration 20

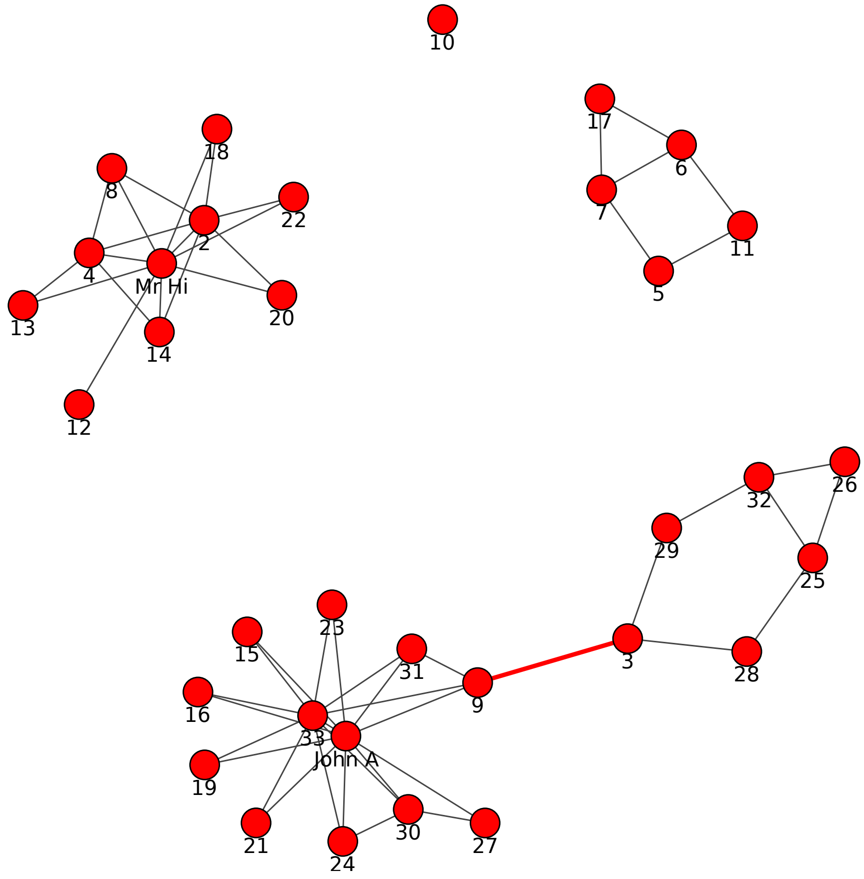

Fig. 2.12. Iteration 21

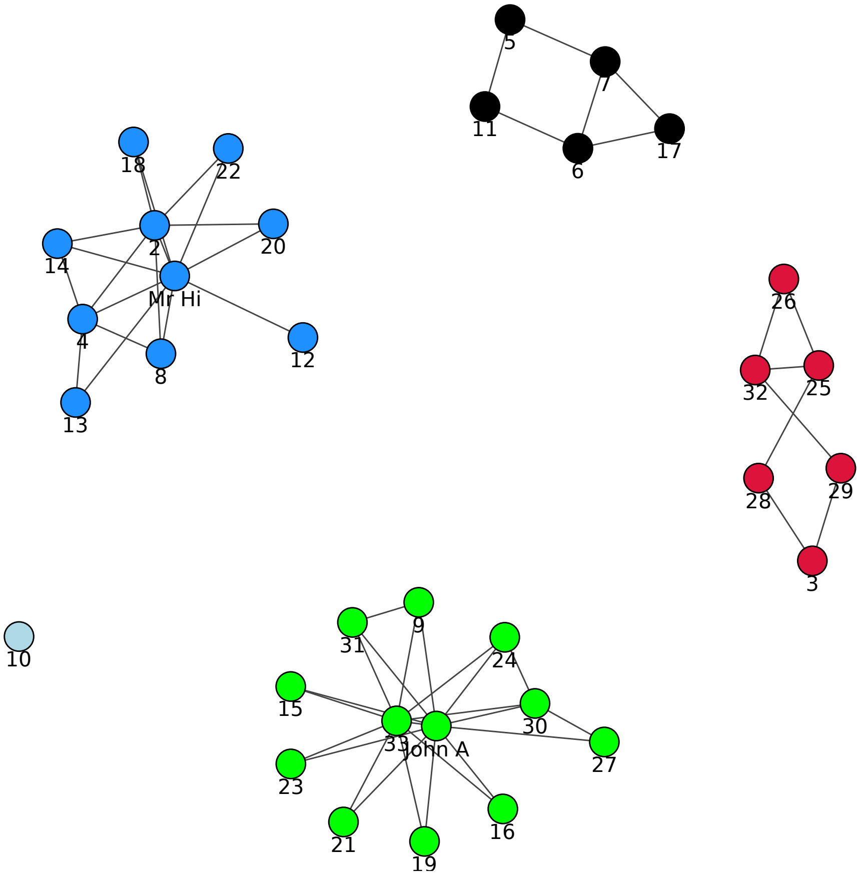

Fig. 2.13. Output of Girvan Newman algorithm which divides the Karate Club Graph divided into 5 groups