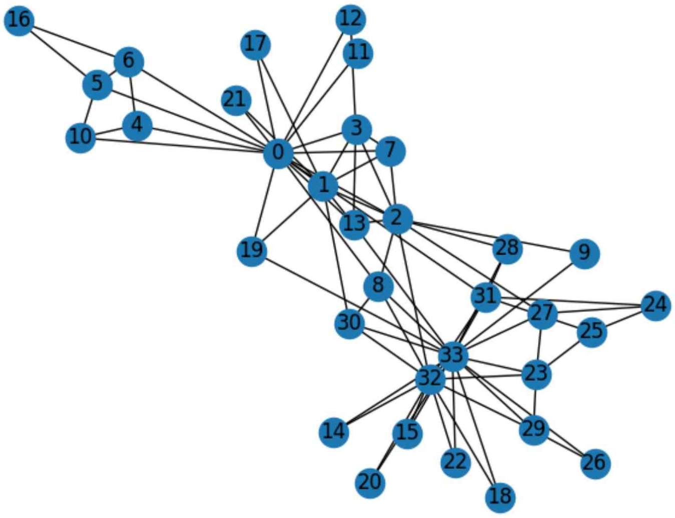
# HW#1

2022711835 Junhee Kwon

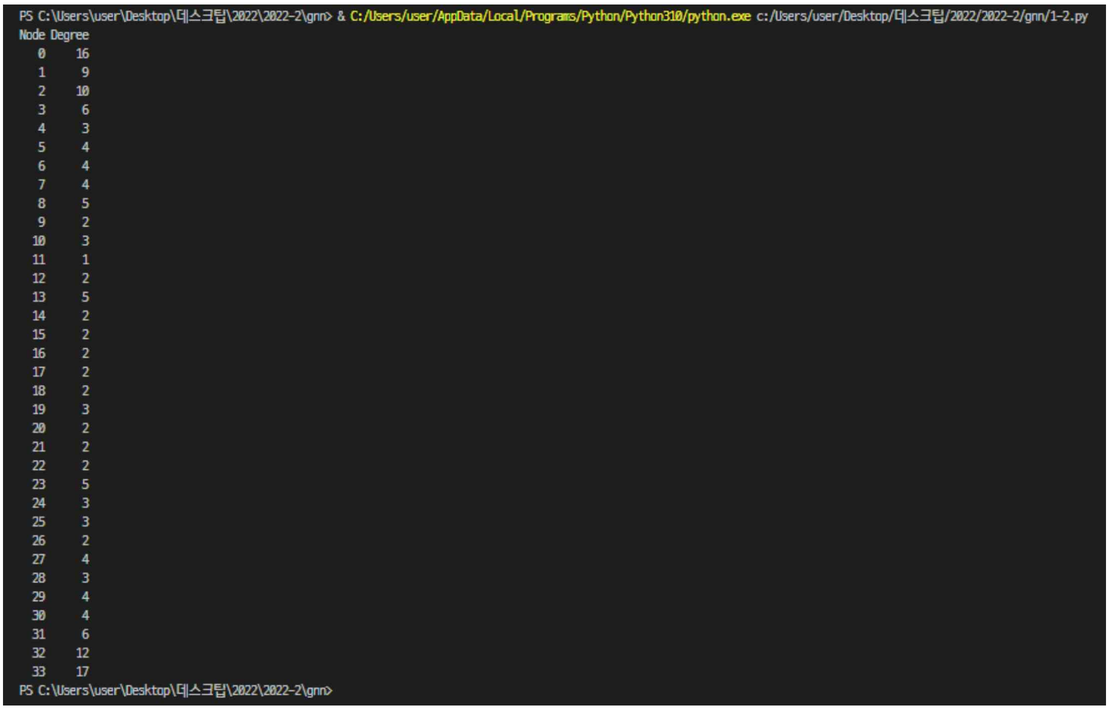
## 1. Data Preparation

Through matplotlib, I drew the graph of the karate club dataset.



## 2. Degree Computation

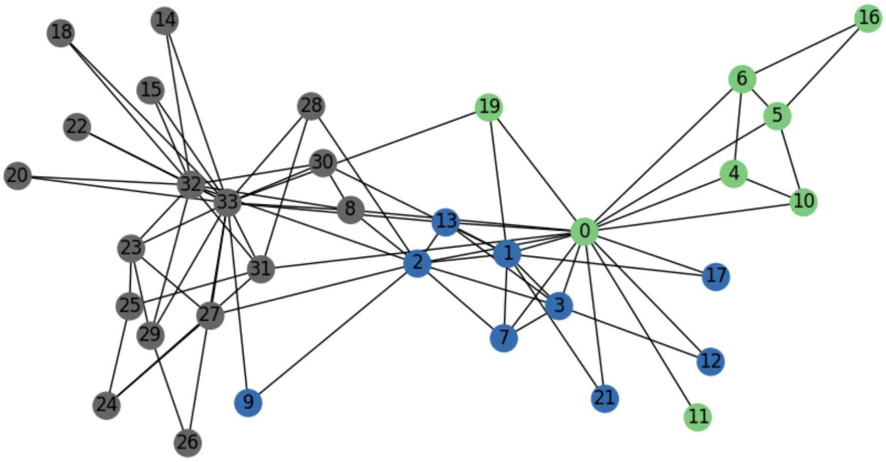
By using the degree method of the network library, I could find the degree information of the nodes.



```
PS C:\Users\user\Desktop\데스크탑\2022\2022-2\gnn> & C:/Users/user/AppData/Local/Programs/Python/Python310/python.exe c:/Users/user/Desktop/데스크탑/2022/2022-2/gnn/1-2.py
Node Degree
   0    16
   1     9
   2    10
   3     6
   4     3
   5     4
   6     4
   7     4
   8     5
   9     2
  10     3
  11     1
  12     2
  13     5
  14     2
  15     2
  16     2
  17     2
  18     2
  19     3
  20     2
  21     2
  22     2
  23     5
  24     3
  25     3
  26     2
  27     4
  28     3
  29     4
  30     4
  31     6
  32    12
  33    17
PS C:\Users\user\Desktop\데스크탑\2022\2022-2\gnn>
```

## 3. Community Detection

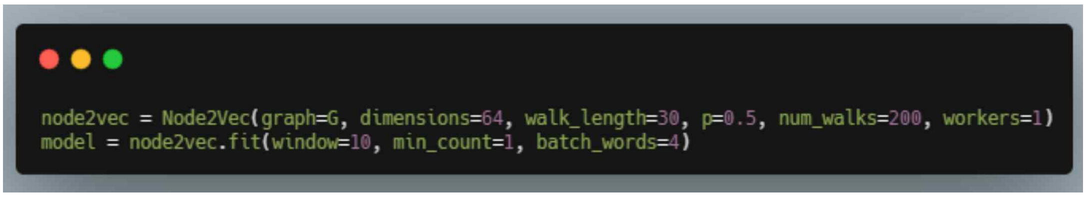
Through the *greedy_modularity_communities()* method of the networks library, I could cluster the nodes in 3 communities.



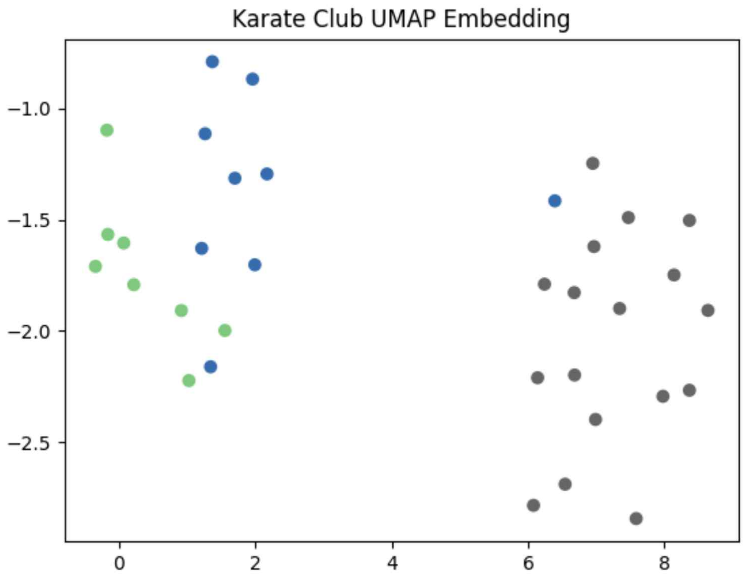
## 4. Learning node2vec with initial parameters

Based on the GitHub - eliorc/node2vec: Implementation of the node2vec algorithm. page, I set the initial values of the node2vec method.

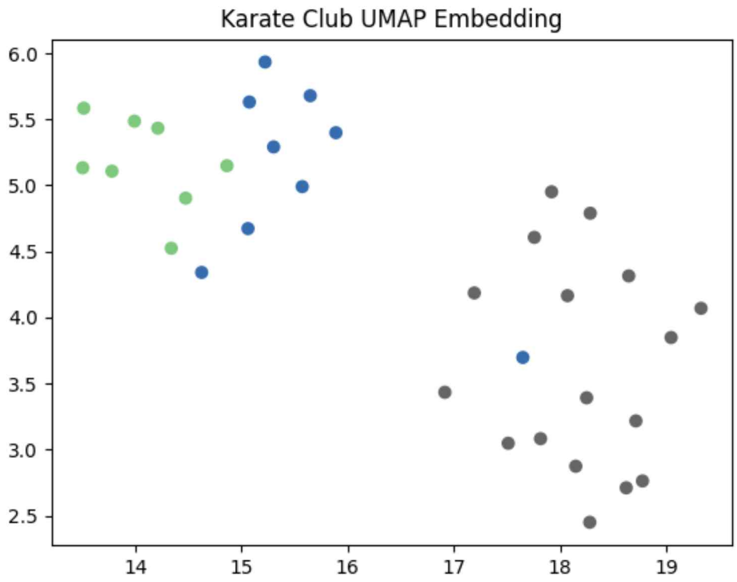
You can check out the node2vec code with the initial values below.



```
node2vec = Node2Vec(graph=G, dimensions=64, walk_length=30, p=0.5, num_walks=200, workers=1)
model = node2vec.fit(window=10, min_count=1, batch_words=4)
```



## 5. Learning node2vec with your best parameters



To make result better than that of initial values, I tried to adjust the parameters.

**Initial parameter**

- p=0.5 (default)
- q=1 (default)
- dimensions=64
- walk-length=30
- num-walks=200
- workers=1 (for Windows)
- windows=10

**Adjusted parameter**

- p= 0.4
- q= 0.8
- dimensions=64
- walk-length=30
- num-walks=200
- workers=1 (for Windows)
- windows=15

1. Values of p and q

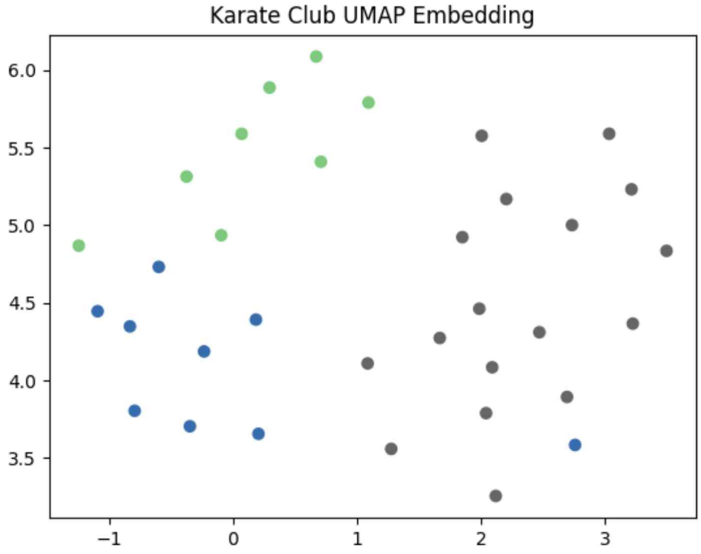
   A. Low p and high q case can explore the graph like BFS that considers the narrow region.

   B. High p and low q case can explore the graph like DFS that considers the broad region.

   C. Low p and high q case shows the lower error than other cases. To find the community, we can use the low p and high q.

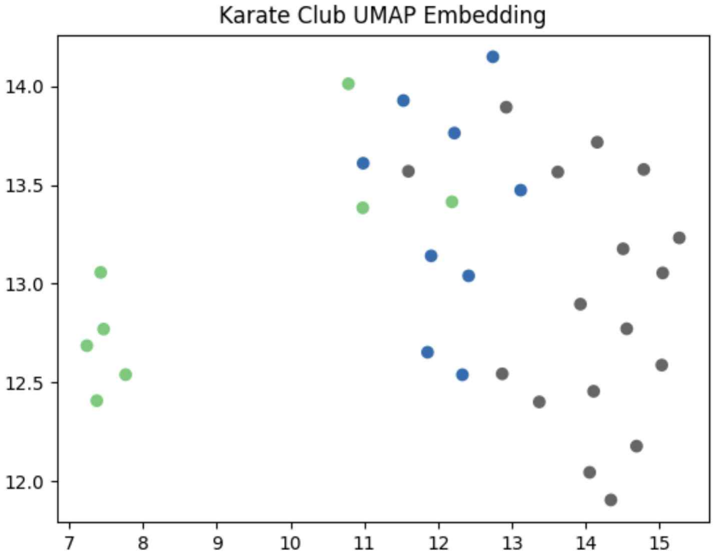
   D. Low p & High q case's graph (Extremely)

   ➔ It searches the nodes near the base node, but ***q*** is too big to find unrelated nodes.



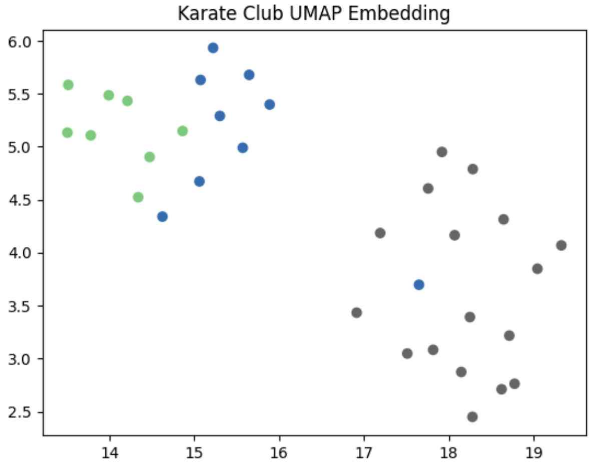
   E. High p & Low q case's graph (Extremely)

   ➔ It cannot distinguish the communities, because this condition searches the nodes based on depth.



F. Moderately small p & large q

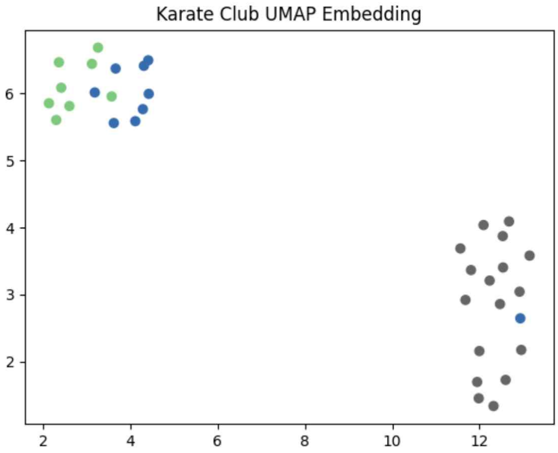
➔ works well



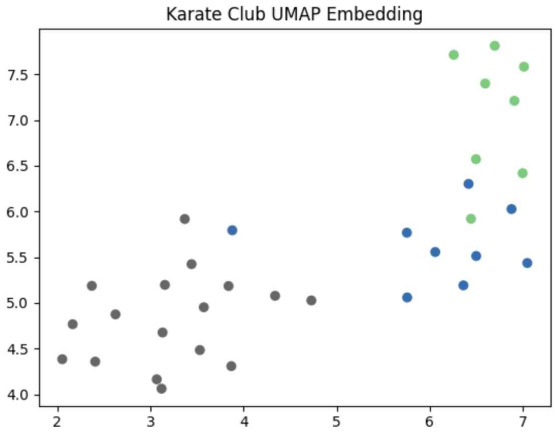
2. For better result, adjusted the other parameters

A. Change window value

➔ from 10 to 15: more clustered, but green nodes and blue nodes are mixed



➔ from 10 to 5: more sparsed



B. Smaller and bigger walk-length made result worse

C. num_walks and dimension also do not have effect to the result.