E3

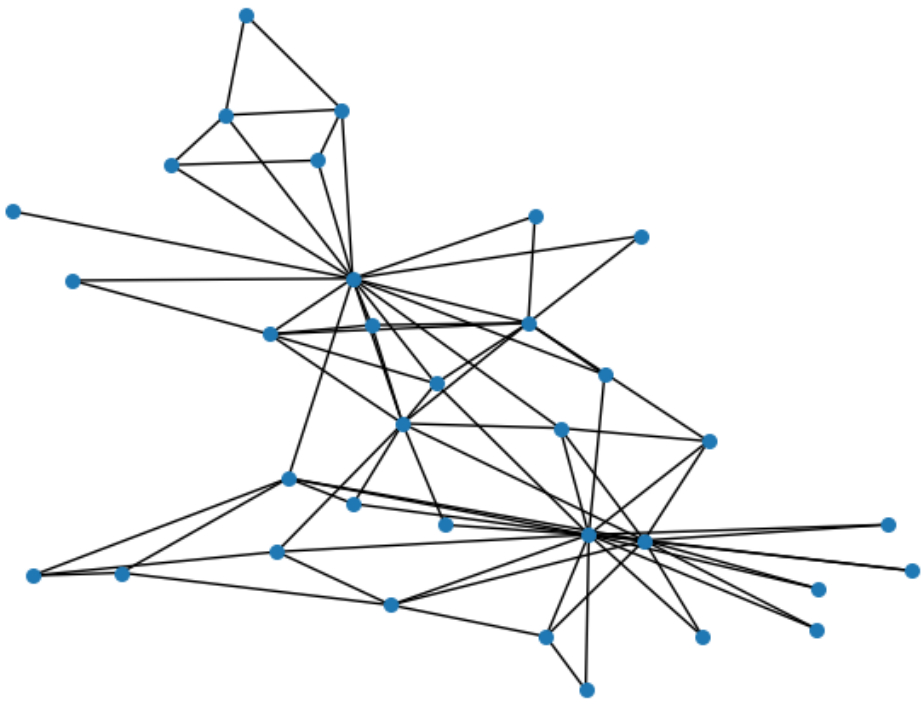
October 17, 2022

[2]:
```
#import the neccessary packages
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

[3]:
```
data = pd.read_csv('zachary.txt', sep=" ", header=None).to_numpy()
A = data[0:34,1:]
print(np.shape(A))
G = nx.from_numpy_matrix(A)
```

```
(34, 34)
```

[4]:
```
nx.draw(G, node_size=30)
```



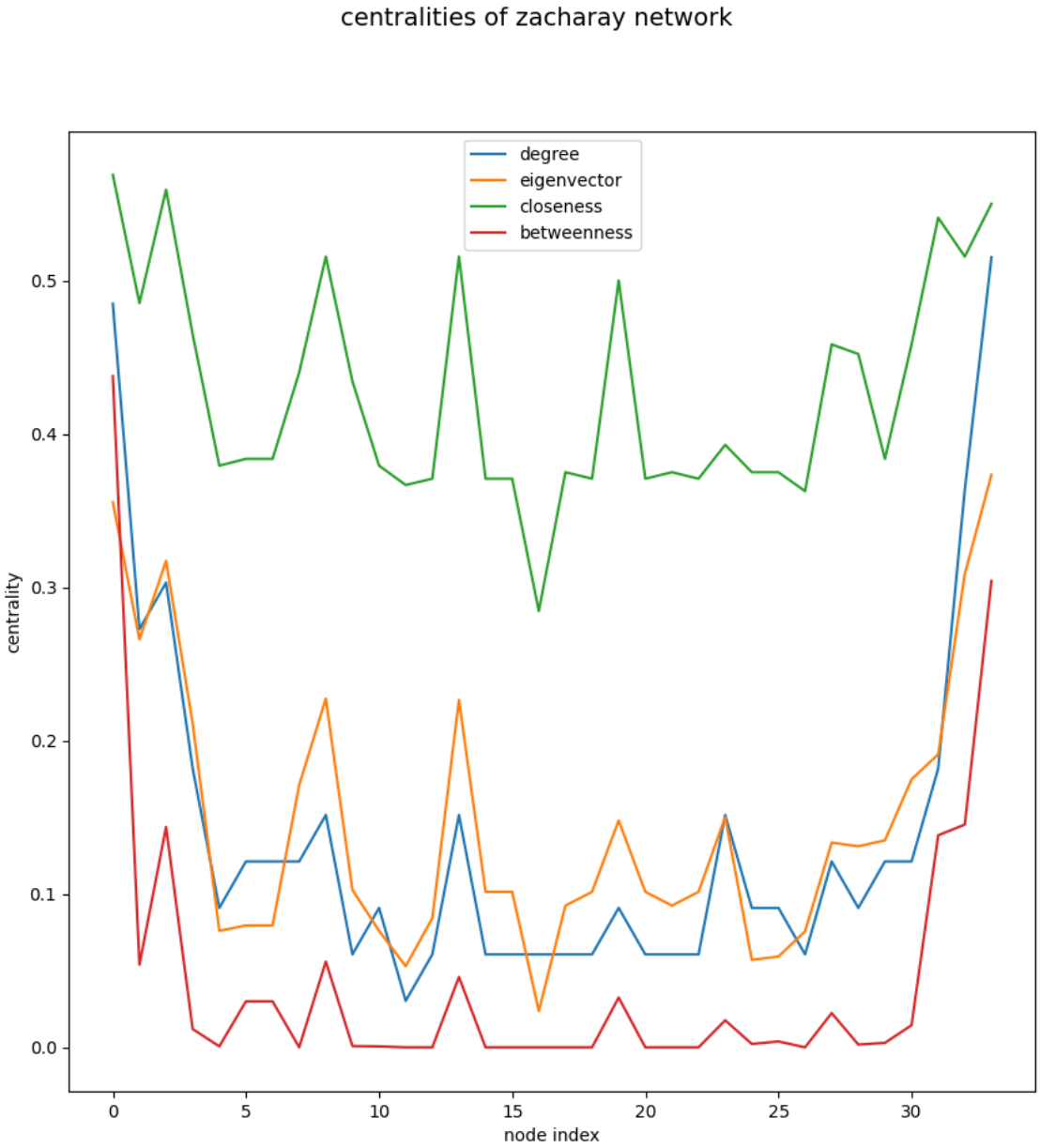
1 Ex 3.1

[5]:
```
dc = nx.degree_centrality(G)
ec = nx.eigenvector_centrality(G)
cc = nx.closeness_centrality(G)
bc = nx.betweenness_centrality(G)
```

[6]:
```
fig, axis = plt.subplots(nrows=1, ncols=1, figsize=(10, 10))
fig.suptitle("centralities of zacharay network", fontsize=14)

axis.plot(dc.keys(), dc.values())
axis.plot(ec.keys(), ec.values())
axis.plot(cc.keys(), cc.values())
axis.plot(bc.keys(), bc.values())
axis.set_xlabel('node index')
axis.set_ylabel('centrality')
axis.legend(['degree', 'eigenvector', 'closeness', 'betweenness'])
```

[6]:
```
<matplotlib.legend.Legend at 0x7ff26fe36a10>
```

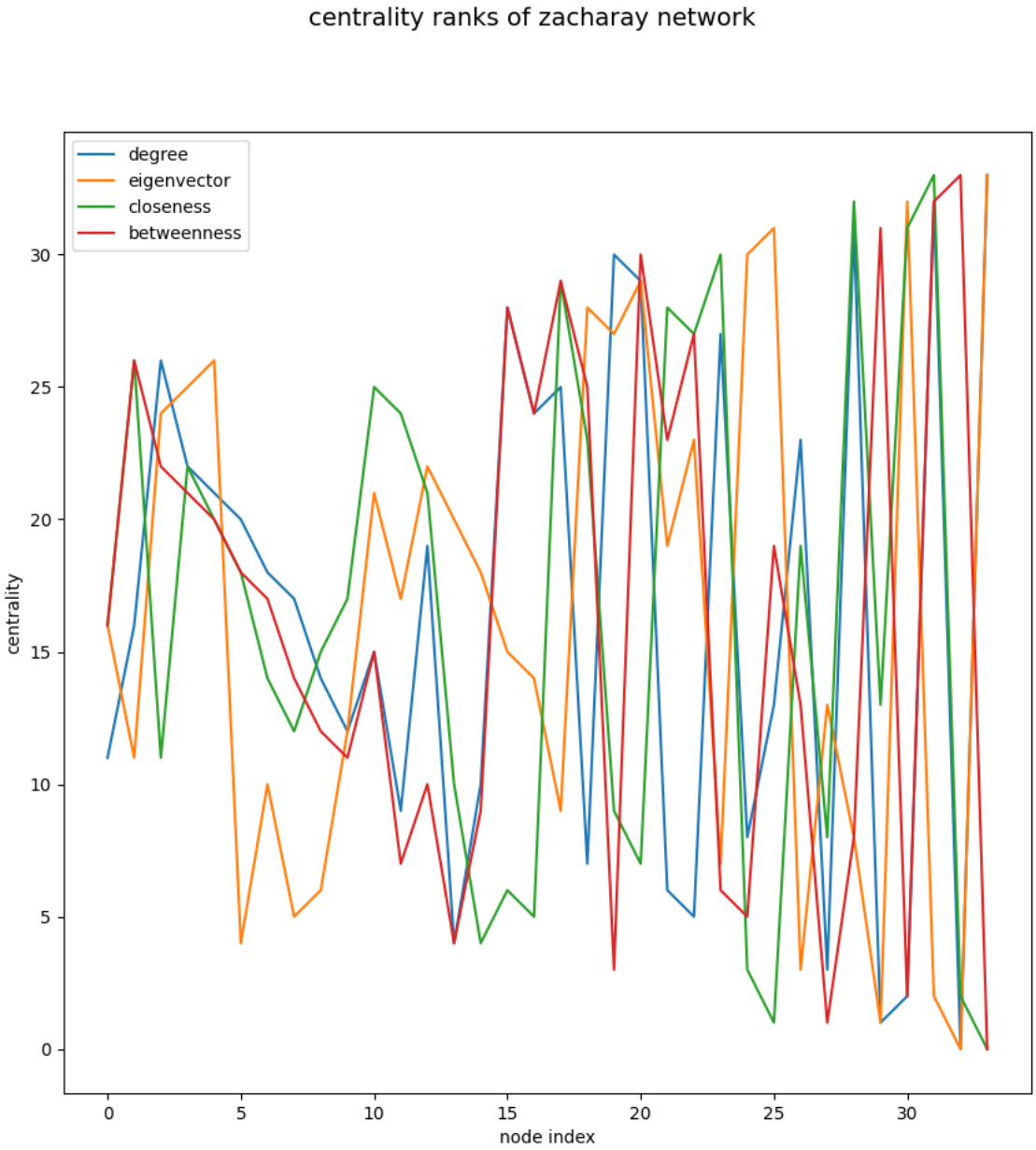


[7]:
```
fig, axis = plt.subplots(nrows=1, ncols=1, figsize=(10, 10))
fig.suptitle("centrality ranks of zacharay network", fontsize=14)

axis.plot(dc.keys(), np.array(list(dc.values())).argsort())
axis.plot(dc.keys(), np.array(list(ec.values())).argsort())
axis.plot(dc.keys(), np.array(list(cc.values())).argsort())
axis.plot(dc.keys(), np.array(list(bc.values())).argsort())
axis.set_xlabel('node index')
axis.set_ylabel('centrality')
```

```
axis.legend(['degree', 'eigenvector', 'closeness', 'betweenness'])
```

[7]: <matplotlib.legend.Legend at 0x7ff26dba72e0>



2 Ex 3.2

[8]:
```
import scipy.stats as stats

centralities = [dc, ec, cc, bc]
corr_matrix = np.zeros((4,4))
for i in range(4):
    for j in range(4):
        pcc = stats.pearsonr(
            list(centralities[i].values()),
            list(centralities[j].values()))

        corr_matrix[i,j] = pcc.statistic

corr_matrix
```

[8]:
```
array([[1.        , 0.91725533, 0.771591  , 0.91464303],
       [0.91725533, 1.        , 0.90461528, 0.80321281],
       [0.771591  , 0.90461528, 1.        , 0.71794456],
       [0.91464303, 0.80321281, 0.71794456, 1.        ]])
```

3 Ex 3.3

[9]:
```
centralities = [dc, ec, cc, bc]
corr_matrix = np.zeros((4,4))
for i in range(4):
    for j in range(4):
        kdt = stats.kendalltau(
            np.array(list(centralities[i].values())).argsort(),
            np.array(list(centralities[j].values())).argsort())
        corr_matrix[i,j] = kdt.correlation

corr_matrix
```

[9]:
```
array([[ 1.        , -0.02673797,  0.0516934 ,  0.09090909],
       [-0.02673797,  1.        , -0.16934046, -0.2228164 ],
       [ 0.0516934 , -0.16934046,  1.        ,  0.09803922],
       [ 0.09090909, -0.2228164 ,  0.09803922,  1.        ]])
```

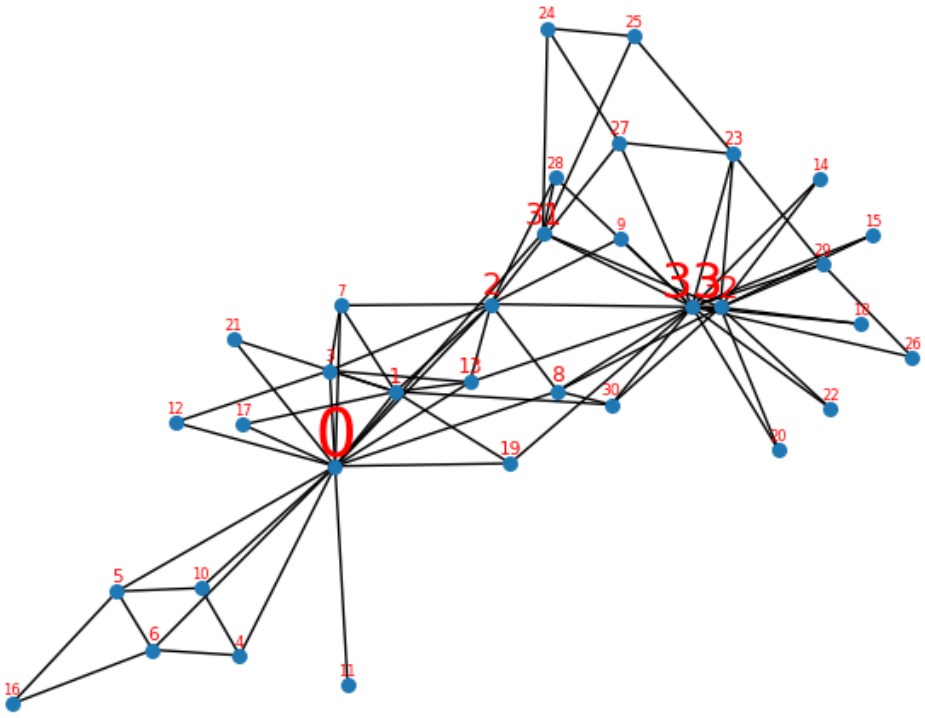
4 Ex 3.4

[40]:
```
pos=nx.spring_layout(G)

nx.draw(G,pos, node_size=30)

for i in range(G.number_of_nodes()):
```

```
    x,y=pos[i]
    plt.text(x,y+0.02,s=str(i),horizontalalignment='center', fontsize=(50*bc[i]␣
↪+ 6), color='red')
```



We can observe thath especially nodes 0 and 33 score very high in terms of all centrality indices. This is also observable in the graph as 0 connects two subcomponents and both 33 and 32 seem to form the center of some sort of star graph.