Graphs Centrality

- The *centrality* of a node in a graph indicates the importance of a node within its given graph or network
- Multiple centrality measures
- Numerical (scalar) value that gives us some information about how well the node connects the network

Different Notions of Centrality (1)

- Betweenness, Degree, Closeness Centrality
- Betweeness Centrality: number of shortest paths in which the node is present
- Degree Centrality: Fraction of nodes that are incident upon a node, same as degree up to scaling

Different Notions of Centrality (2)

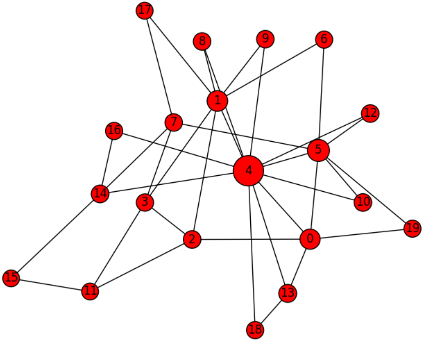
- Closeness Centrality: compute shortest path distance to all other nodes in the network, average the results and divide by the maximum distance
- Eigenvector Centrality: How important is the node from the perspective of random crawler, how often does a random crawler return to this node
    - Markov Chains: static or steady-state probability of visiting a node, eigenvector of the adjacency matrix with the greatest eigenvalue

Centrality Measures implemented in `networkx`

- Centrality measures are implemented in `networkx` as shown below
- List of the values in a python dictionary can be accessed

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np

G = nx.krackhardt_kite_graph()
#G = nx.erdos_renyi_graph(30, 0.1)

print(nx.degree_histogram(G))

print(nx.betweenness_centrality(G))
print(nx.degree_centrality(G))
print(nx.closeness_centrality(G))
print(nx.eigenvector_centrality(G))

D=nx.eigenvector_centrality(G)
D=list(D.values())
```

Visualizing Centrality



- An example where we weight the nodes visually with a value corresponding to their degree

In-Class Exercise

- As shown in the sample script, compute the four measures of node centrality for a graph G
- Draw the graph where the size of any node is proportional to its degree
  - Note that node size can be controlled with the argument `node_size=`
- Examples for each of the following types of graphs: Barabasi-Albert Graph, Krackhardt Kite Graph, Erdos Renyi Graph