Problem4

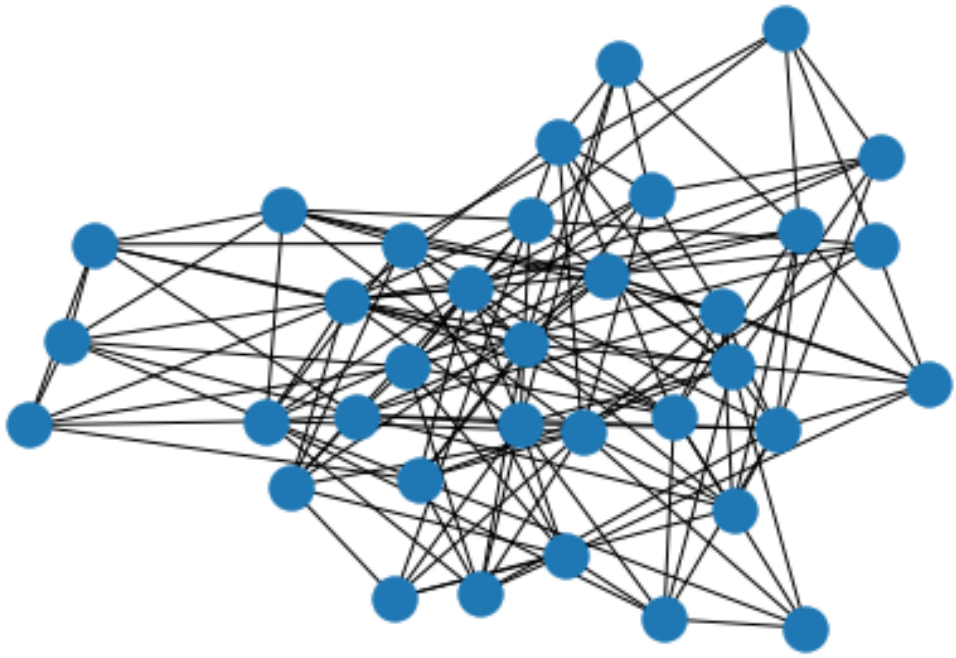
January 25, 2023

```
In [121]: import networkx as nx
          import matplotlib.pyplot as plt
          import numpy as np
          import pickle
```

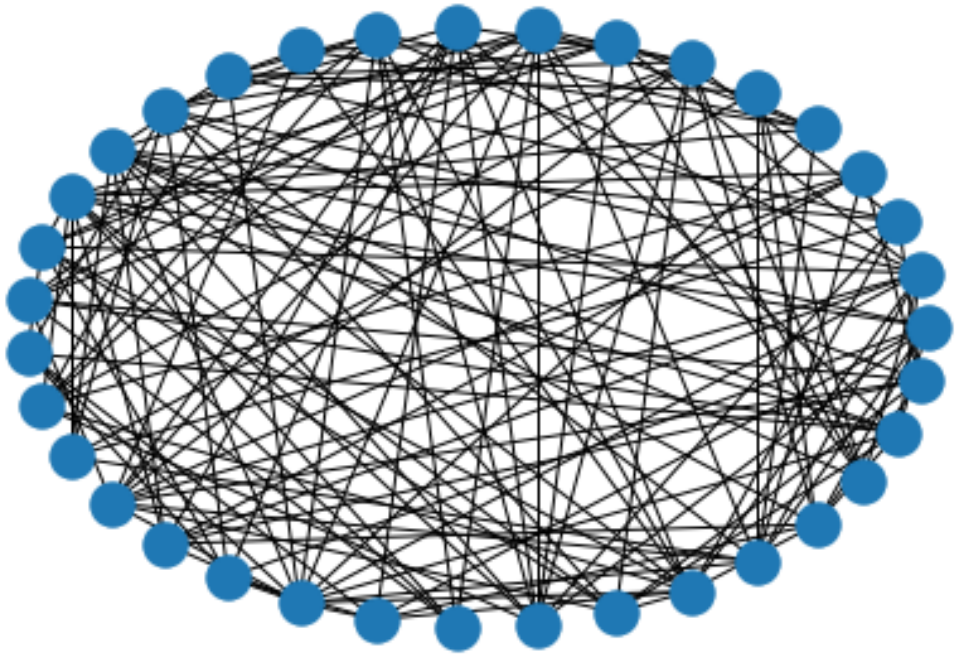
1 A.

```
In [122]: G = nx.Graph()
          n = 35
          p = 0.25
          for i in range(n):
              G.add_node(i)
          for i in range(n):
              for j in range(i+1, n, 1):
                  if np.random.random() <= p:
                      G.add_edge(i, j)
```

```
In [123]: nx.draw(G)
```



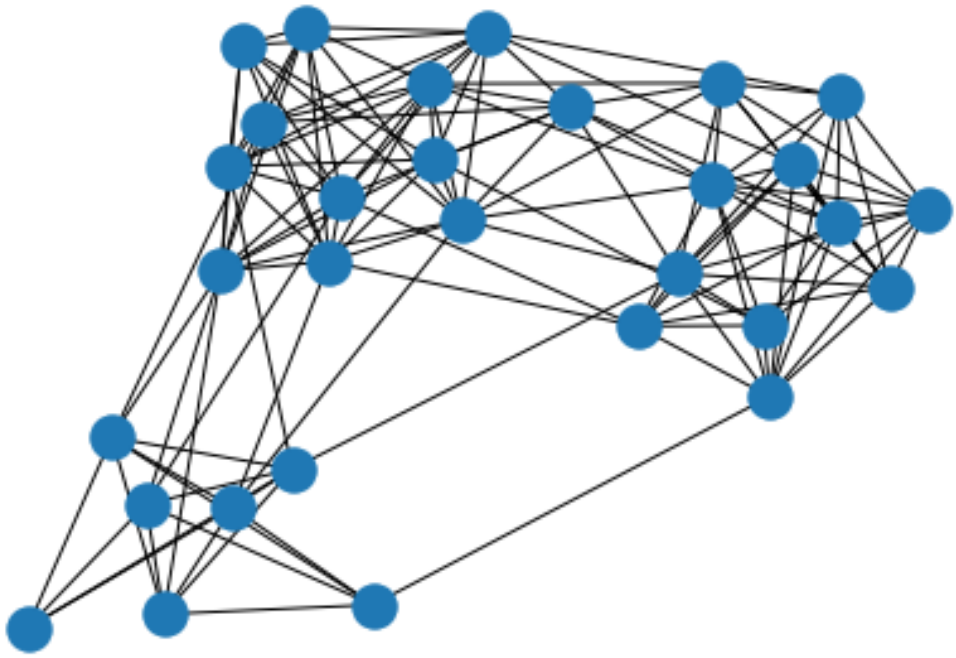
In [124]: nx.draw_circular(G)



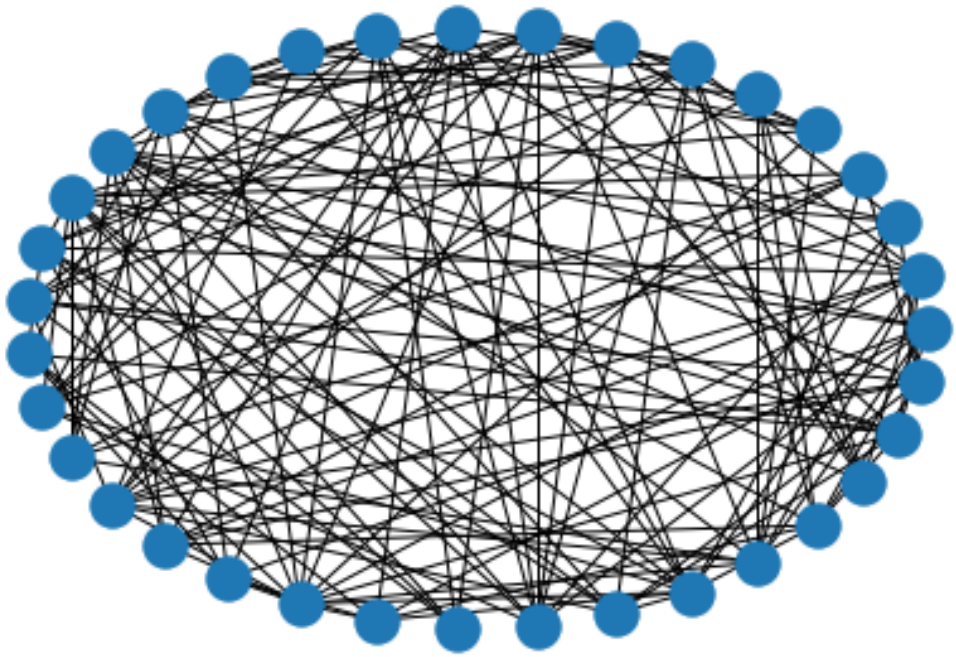
2 B

In [98]:
```
n = 30
k = 3
A = 0.7
B = 0.1
W = [[A, B, B], [B, A, B], [B, B, A]]

G = nx.Graph()
labels = [0, 1, 2]

for i in range(n):
    label = np.random.choice(labels)
    G.add_node(str(i) + " " + str(label))

nodes = list(G.nodes)
for i in range(n):
    for j in range(i + 1, n, 1):
        node1 = nodes[i]
        node2 = nodes[j]
        label1 = node1.split(" ")[1]
        label2 = node2.split(" ")[1]
        p = W[int(label1)][int(label2)]
        if np.random.random() < p:
            G.add_edge(node1, node2)
```

In [89]:
```
nx.draw(G)
```



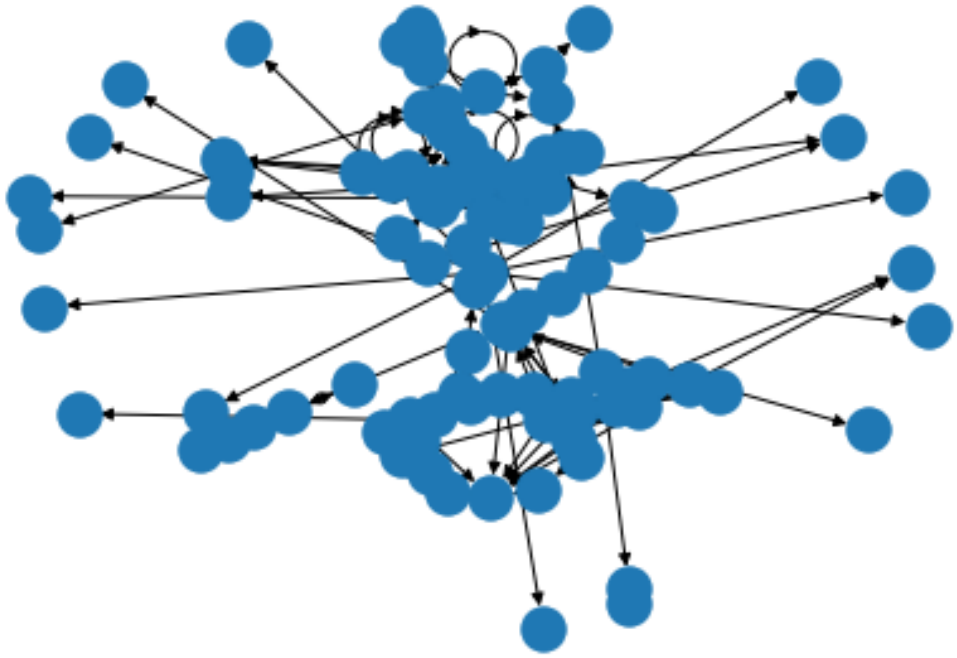
In [131]: nx.draw_circular(G)



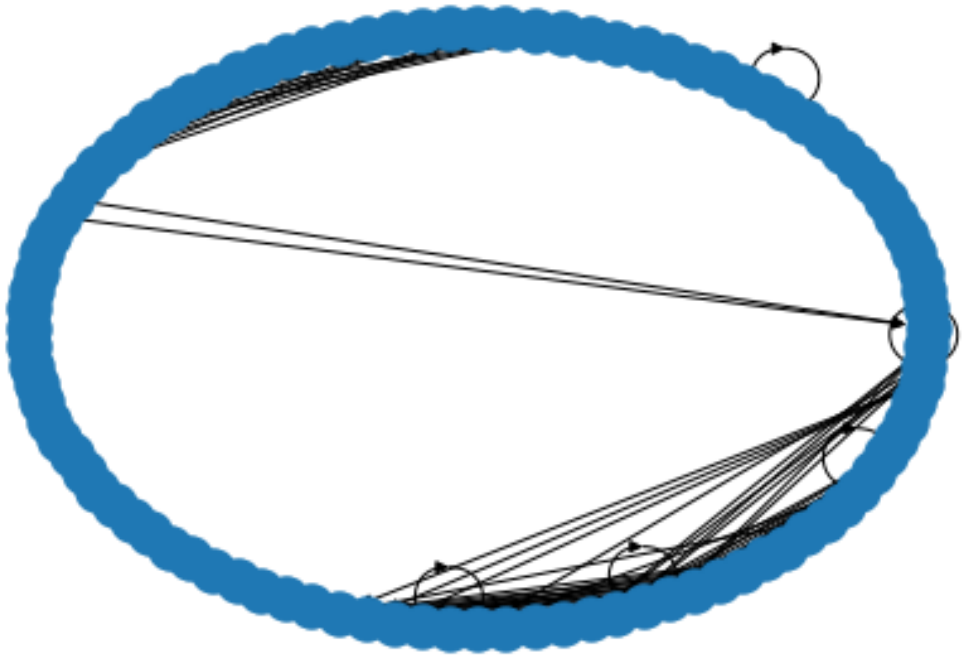
3 C

```
In [100]: with open("web_graph_full.pickle", "rb") as graph_file:
              web_graph_100 = pickle.load(graph_file)
          web_graph_100.remove_nodes_from(list(web_graph_100.nodes)[100:])
```

```
In [101]: nx.draw(web_graph_100)
```

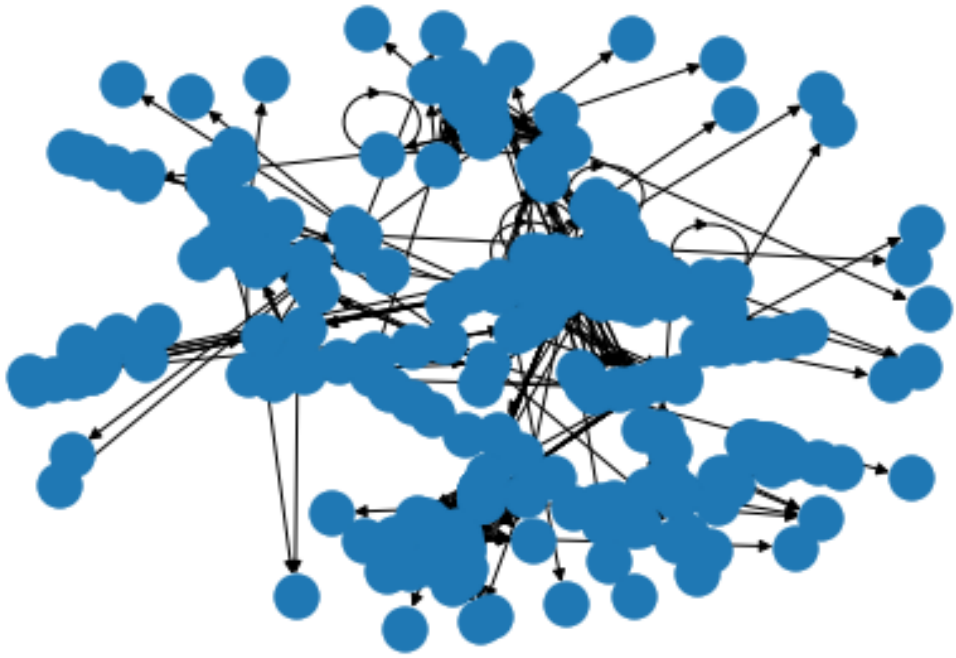


```
In [126]: nx.draw_circular(web_graph_100)
```

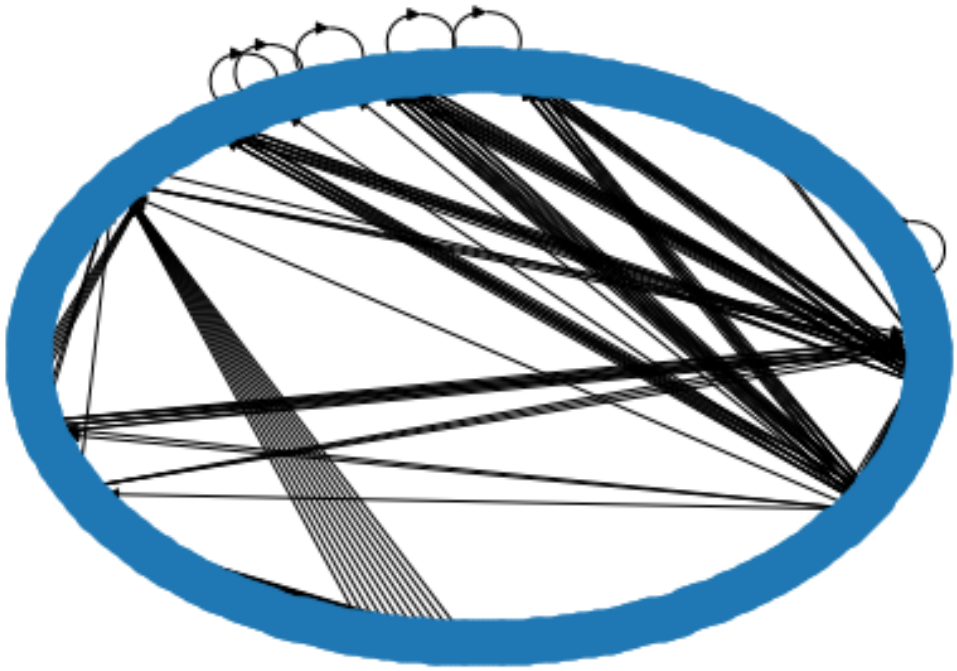


```
In [104]: with open("web_graph_full.pickle", "rb") as graph_file:
              web_graph_300 = pickle.load(graph_file)
          web_graph_300.remove_nodes_from(list(web_graph_300.nodes)[300:])
```

```
In [128]: nx.draw(web_graph_300)
```



In [129]: nx.draw_circular(web_graph_300)



The default draw is the spring visualization, which tends to show clusters well by increasing spring force on nodes with lots of edges. The nodes look very clustered there, while the circular visualization lets us see the density in edges between clusters pretty well, and the presence of smaller clusters better. I used these 2 visualizations for all of these.

For the erdos graph, we see that it is truly random. There are no outstanding clusters and the edge density seems pretty even in the circular graph as well.

for the SBM graph, we see the 3 clusters we should see with the spring graph, but we see that the distribution of edge densities is pretty random still.

In both the web graphs, we see some big distinct clusters with the spring visualization, but we also see pretty numerous smaller clusters with large edge densities, or at the very least can see the connections between highly connected nodes such as home pages.