Assignment 5

March 1, 2021

Name: Craig Fox

TUID: 915781095

1 Problem 1

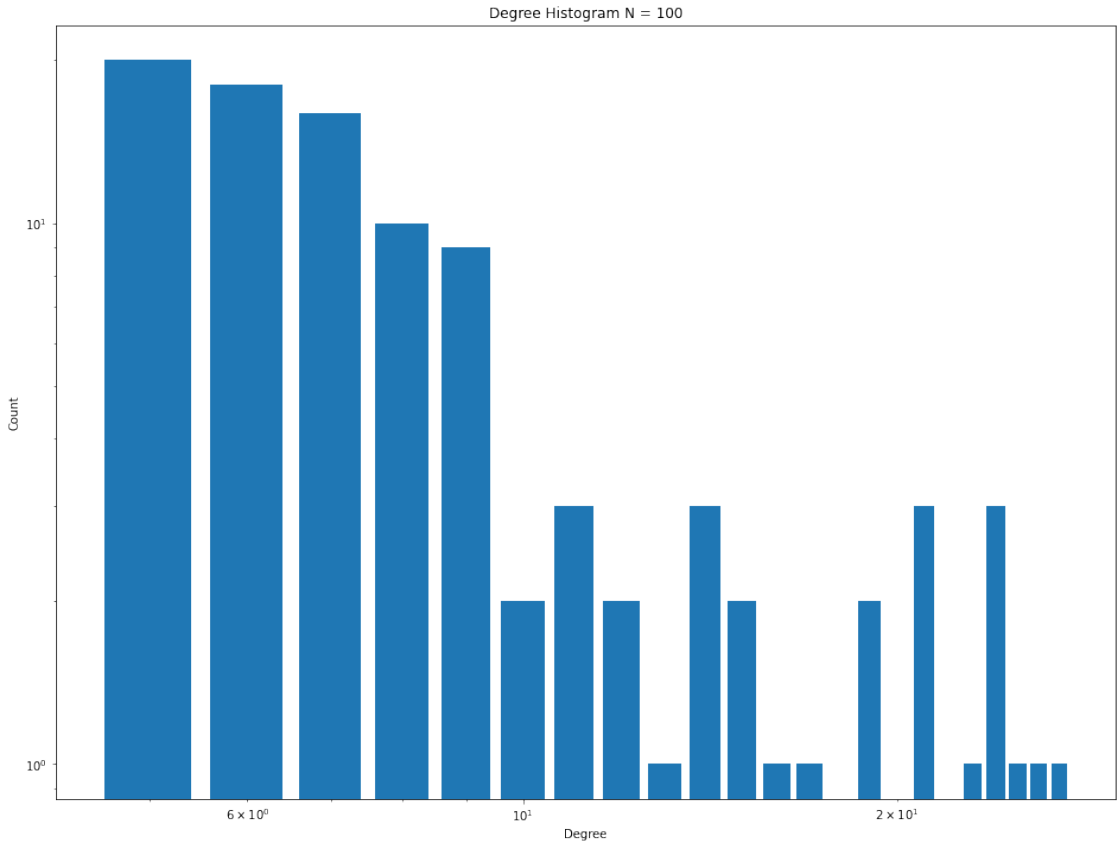
There is only one way for D to enter without it being in an unbalanced triangle. If D has positive relations with A, B, and C, then every triangle involving D will have one negative and two positives which is a valid combo. The whole network is unbalanced because A, B, and C form an unbalanced triangle, but D would be balanced.

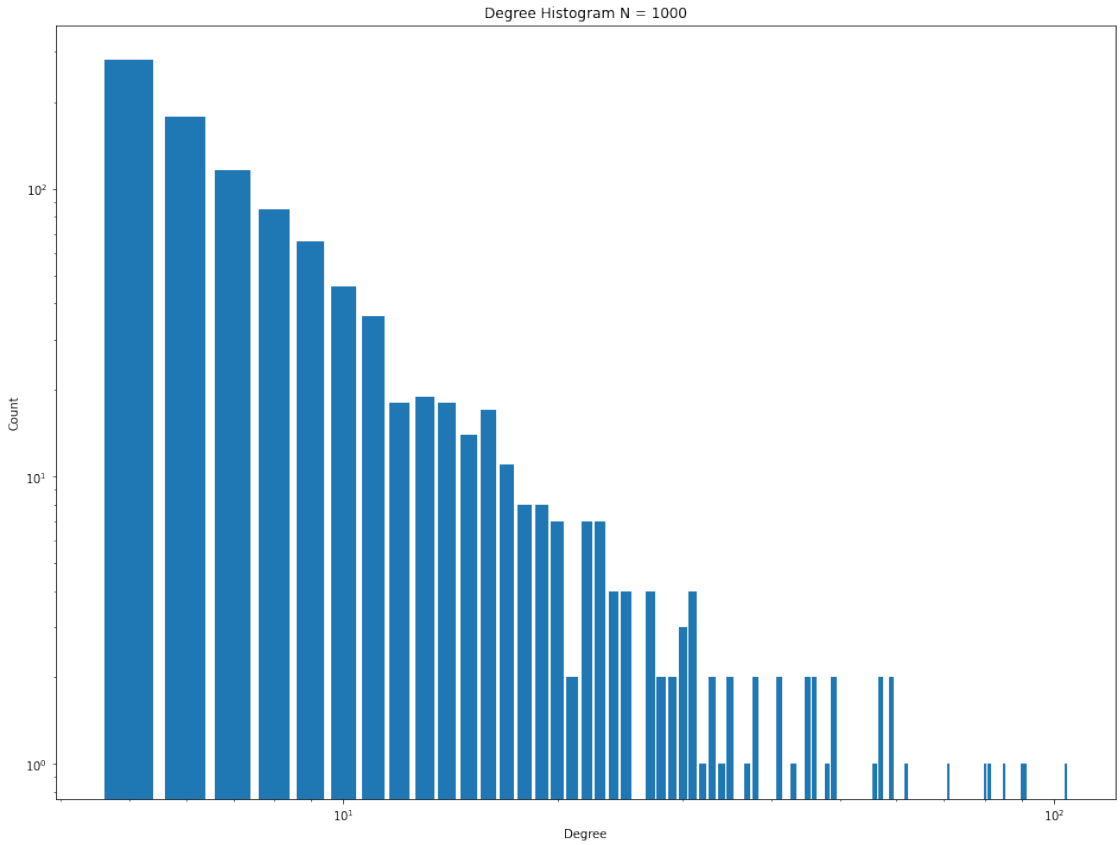
2 Problem 2

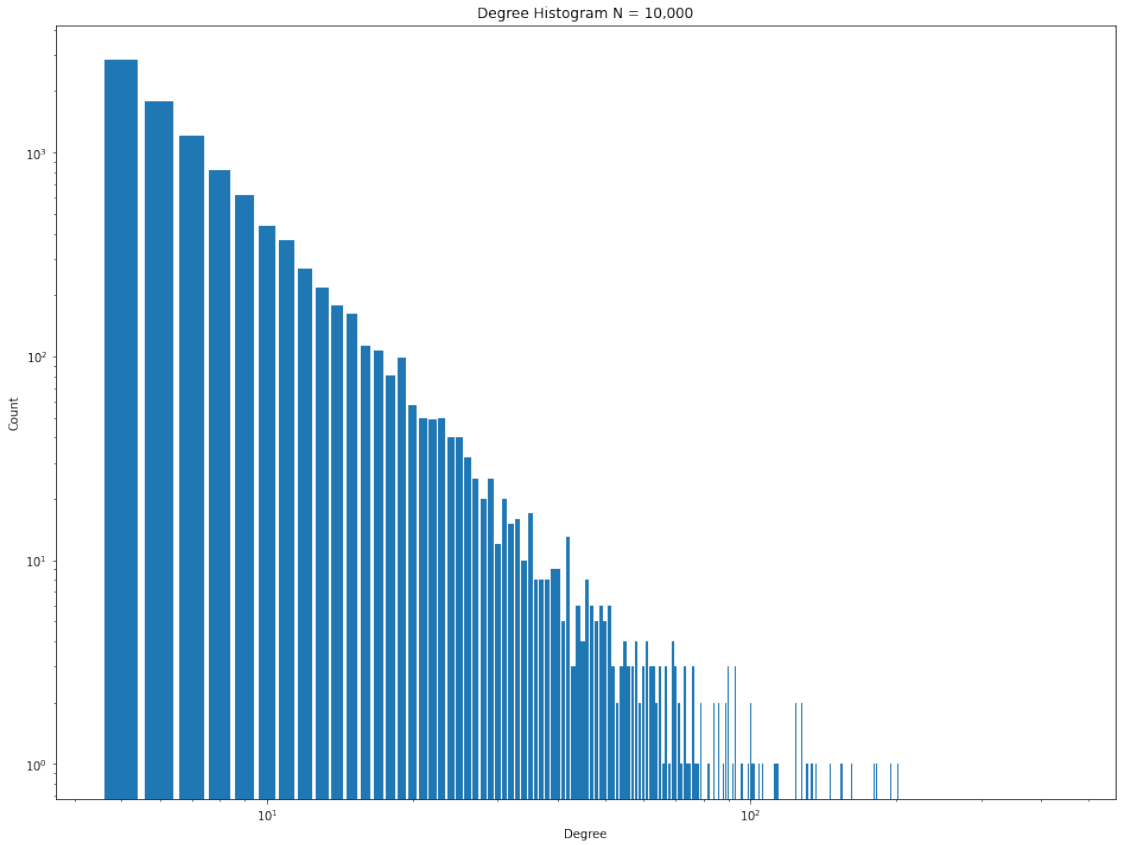
[36]:
```
import networkx as nx
import random
import matplotlib.pyplot as plt
import collections

m=5

print('Part a')

G100 = nx.barabasi_albert_graph(100, m)
degree_sequence = sorted([d for n, d in G100.degree()], reverse=True)
degreeCount = collections.Counter(degree_sequence)
deg, cnt = zip(*degreeCount.items())

plt.figure(figsize=(16, 12))
plt.bar(deg, cnt)

plt.title("Degree Histogram N = 100")
plt.ylabel("Count")
plt.xlabel("Degree")
plt.xscale('log')
plt.yscale('log')
plt.show()

G1000 = nx.barabasi_albert_graph(1000, m)
```

```
degree_sequence = sorted([d for n, d in G1000.degree()], reverse=True)
degreeCount = collections.Counter(degree_sequence)
deg, cnt = zip(*degreeCount.items())

plt.figure(figsize=(16, 12))
plt.bar(deg, cnt)

plt.title("Degree Histogram N = 1000")
plt.ylabel("Count")
plt.xlabel("Degree")
plt.xscale('log')
plt.yscale('log')
plt.show()

G10000 = nx.barabasi_albert_graph(10000, m)
degree_sequence = sorted([d for n, d in G10000.degree()], reverse=True)
degreeCount = collections.Counter(degree_sequence)
deg, cnt = zip(*degreeCount.items())

plt.figure(figsize=(16, 12))
plt.bar(deg, cnt)

plt.title("Degree Histogram N = 10,000")
plt.ylabel("Count")
plt.xlabel("Degree")
plt.xscale('log')
plt.yscale('log')
plt.show()
```

Part a

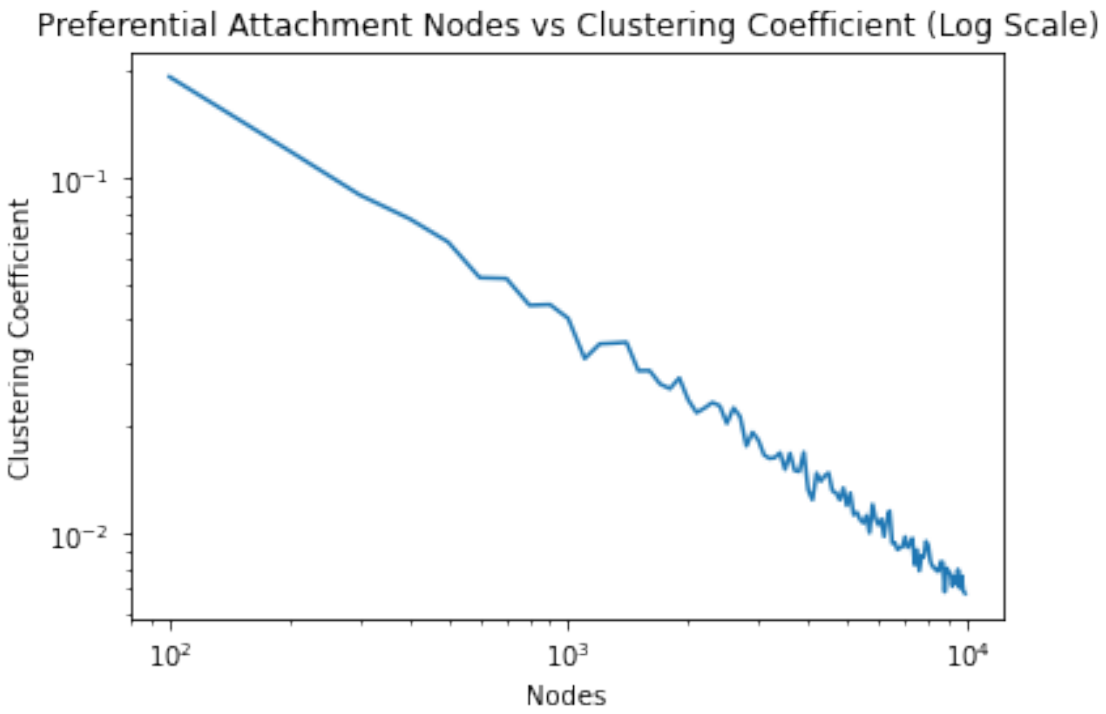

Degree Histogram N = 100
Count
10^1
10^0
6×10^0
10^1
2×10^1
Degree


Degree Histogram N = 1000
Count
10^2
10^1
10^0
10^1
10^2
Degree



[31]:
```
print('Part b')
X = []
Y = []

for x in range(100,10000,100):
    X.append(x)
    G = nx.barabasi_albert_graph(x, m)
    Y.append(nx.cluster.average_clustering(G))
plt.title("Preferential Attachment Nodes vs Clustering Coefficient (Log Scale)")
plt.ylabel("Clustering Coefficient")
plt.xlabel("Nodes")
plt.xscale('log')
plt.yscale('log')
plt.plot(X, Y)
print('The Clustering Coefficient is inversely proportional to the number of␣
 ↪nodes as shown by the graph')
```

```
Part b
The Clustering Coefficient is inversely proportional to the number of nodes as
shown by the graph
```



[34]:
```
print('Part c')
print('An initial node has a degree of ' + str(G10000.degree(0)))
print('The t=100 node has a degree of ' + str(G10000.degree(100)))
print('The t=1000 node has a degree of ' + str(G10000.degree(1000)))
print('The t=5000 node has a degree of ' + str(G10000.degree(5000)))
```

```
An initial node has a degree of 168
The t=100 node has a degree of 98
The t=1000 node has a degree of 9
The t=5000 node has a degree of 5
```

3 Problem 3

I would be interested in studying 1. Voting Networks, 2. Antisocial behavior on the web, or 3. Industrial applications of information network.