HOMEWORK #2 – DHRUV SINGH – A20541901

1. RANDOM GRAPH MODEL
   - Original Table

Table 4.4: A Comparison between Real-World Networks and Simulated Random Graphs. In this table, *C* denotes the average clustering coefficient. The last two columns show the average path length and the clustering coefficient for the random graph simulated for the real-world network. Note that average path lengths are modeled properly, whereas the clustering coefficient is underestimated

	Original Network				**Simulated Random Graph**	
Network	*Size*	*Average Degree*	*Average Path Length*	*C*	*Average Path Length*	*C*
Film Actors	225,226	61	3.65	0.79	2.99	0.00027
Medline Coauthorship	1,520,251	18.1	4.6	0.56	4.91	1.8×10^{-4}
E.Coli	282	7.35	2.9	0.32	3.04	0.026
C.Elegans	282	14	2.65	0.28	2.25	0.05

   - My Results

	Average Path Length	C
E. Coil	3.06259306933192	0.026487362125659976
C. Elegans	2.4169001287196186	0.05118802323951168

   - Comparison Table

	Original Network		**Simulated random graph**		**My python code implementation**	
	Average Path Length	C (Clustering Coefficient)	Average Path Length	C (Clustering Coefficient)	Average Path Length	C (Clustering Coefficient)
E.Coli	2.9	0.32	3.04	0.026	3.0625	0.02648
C.Elegans	2.65	0.28	2.25	0.05	2.4169	0.05118

   - Code
     - **E.Coil**

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

# Given data
n = 282
avg_degree = 7.35
avg_path_length_target = 2.9
C_target = 0.32

# Calculate probability for Erdős-Rényi model
p = avg_degree / (n - 1)

# Simulate Erdős-Rényi model
random_graph = nx.erdos_renyi_graph(n, p)
```

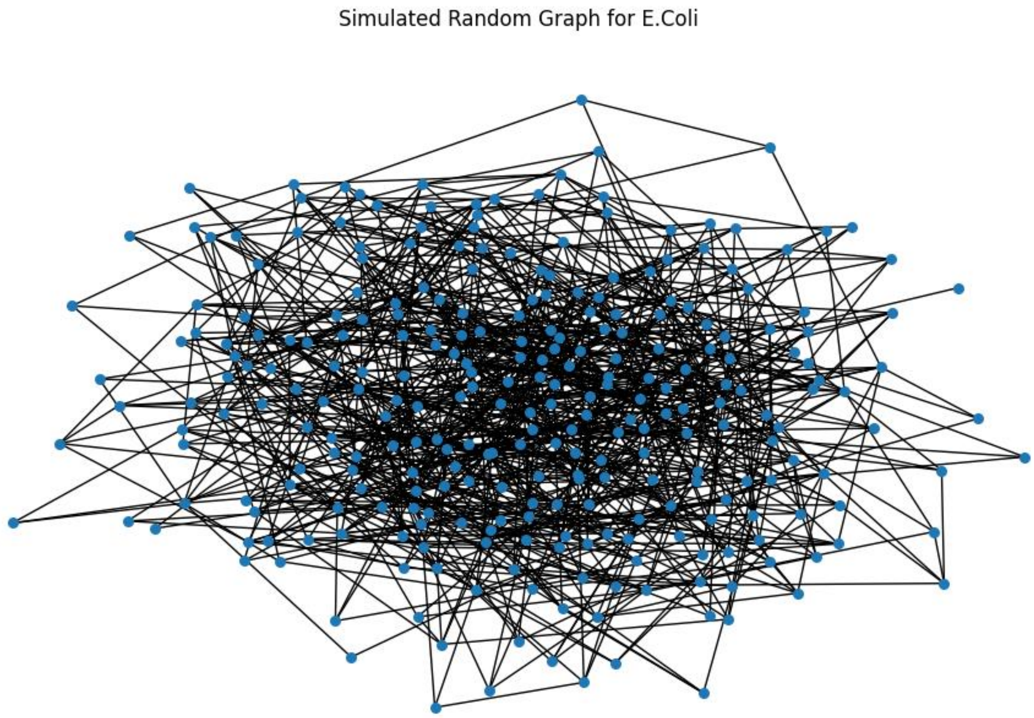
```python
# Calculate average path length
avg_path_length_simulated = nx.average_shortest_path_length(random_graph)

# Calculate clustering coefficient
clustering_coefficient_simulated = nx.average_clustering(random_graph)

# Visualize the graph
plt.figure(figsize=(10, 6))
pos = nx.spring_layout(random_graph)  # Positioning of nodes
nx.draw(random_graph, pos, with_labels=False, node_size=30)
plt.title('Simulated Random Graph for E.Coli')
plt.show()

# Print results
print("Simulated average path length:", avg_path_length_simulated)
print("Simulated clustering coefficient:",
clustering_coefficient_simulated)
```

- C. Elegans

```python
import networkx as nx
import matplotlib.pyplot as plt

# Given data
n = 282
avg_degree = 14
avg_path_length_target = 2.65
C_target = 0.28

# Calculate probability for Erdős-Rényi model
p = avg_degree / (n - 1)

# Simulate Erdős-Rényi model
random_graph = nx.erdos_renyi_graph(n, p)

# Calculate average path length
avg_path_length_simulated =
nx.average_shortest_path_length(random_graph)

# Calculate clustering coefficient
clustering_coefficient_simulated =
nx.average_clustering(random_graph)

# Visualize the graph
plt.figure(figsize=(10, 6))
pos = nx.spring_layout(random_graph)  # Positioning of nodes
```

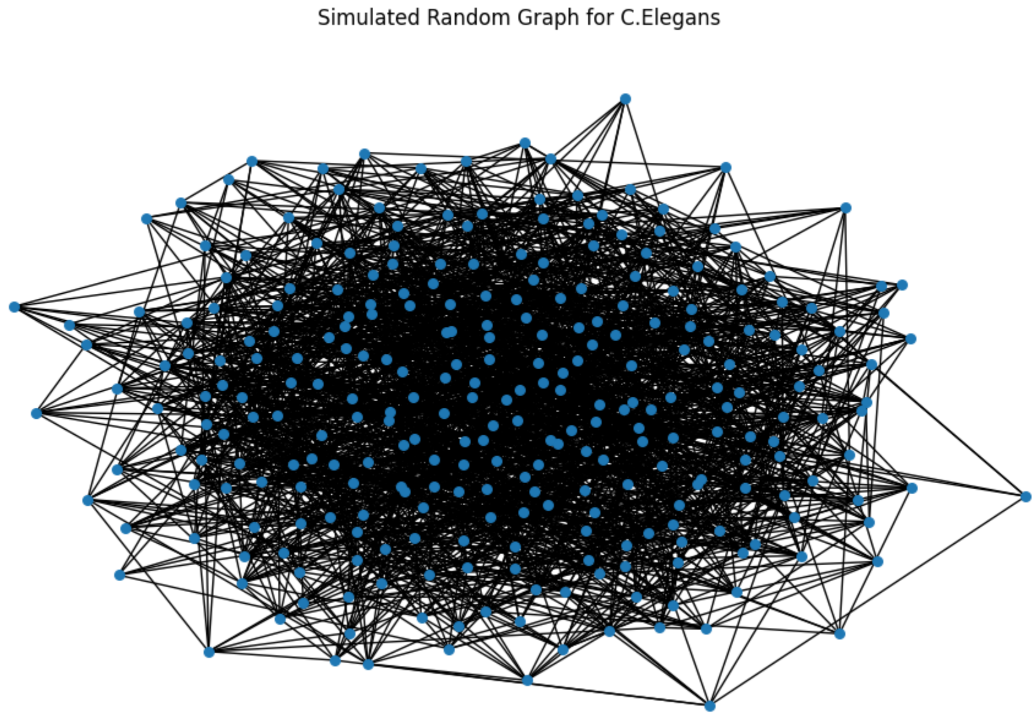
```
nx.draw(random_graph, pos, with_labels=False, node_size=30)
plt.title('Simulated Random Graph for C.Elegans')
plt.show()

# Print results
print("Simulated average path length:", avg_path_length_simulated)
print("Simulated clustering coefficient:",
clustering_coefficient_simulated)
```

- Result
    - E.Coil



Simulated average path length: 3.06259306933192
Simulated clustering coefficient: 0.026487362125659976

- C. Elegans



Simulated average path length: 2.4169001287196186
Simulated clustering coefficient: 0.05118802323951168

- Analysis
  - E.Coil

Average Path Length:
The average path length of my network (3.06259306933192) is closer to the simulated random graph (3.04) than the original network (2.9). This suggests that the structure of your network might be more similar to a random graph in terms of path lengths.

Clustering Coefficient (C):
Both the simulated random graph and my network have similar clustering coefficients (0.026 and 0.026487362125659976 respectively), which are much lower than the clustering coefficient of the original network (0.32). This implies that both your network and the simulated random graph have low clustering, indicating a lack of clustering or local density of connections compared to the original network.

- C.Elegans

Average Path Length:
Your network's average path length (2.4169001287196186) is closer to the original network (2.65) than the simulated random graph (2.25). This suggests that your network might have a structure more similar to the original network in terms of path lengths.
Clustering Coefficient (C):

Both your network and the simulated random graph have similar clustering coefficients (0.051188023239511 68 and 0.05 respectively), which are lower than the clustering coefficient of the original network (0.32). This indicates that both your network and the simulated random graph have lower levels of clustering compared to the original network.

2. SMALL WORLD MODEL

- Original Table

Table 4.5: A Comparison between Real-World Networks and Simulated Graphs Using the Small-World Model. In this table *C* denotes the average clustering coefficient. The last two columns show the average path length and the clustering coefficient for the small-world graph simulated for the real-world network. Both average path lengths and clustering coefficients are modeled properly

	Original Network				**Simulated Graph**	
Network	*Size*	*Average Degree*	*Average Path Length*	*C*	*Average Path Length*	*C*
Film Actors	225,226	61	3.65	0.79	4.2	0.73
Medline Coauthorship	1,520,251	18.1	4.6	0.56	5.1	0.52
E.Coli	282	7.35	2.9	0.32	4.46	0.31
C.Elegans	282	14	2.65	0.28	3.49	0.37

- My Results

	Average Path Length	C
E. Coil	4.190959339744075	0.34807778903523606
C. Elegans	3.282855051614043	0.5959205861583833

- Comparison Table

	Original Network		**Simulated random graph**		**My python code implementation**	
	Average Path Length	C (Clustering Coefficient)	Average Path Length	C (Clustering Coefficient)	Average Path Length	C (Clustering Coefficient)
E.Coli	2.9	0.32	4.46	0.31	4.19095	0.34807
C.Elegans	2.65	0.28	3.49	0.37	3.28285	0.59592

- Code
  - E.Coil

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

# Given data
n = 282
avg_degree = 7.35
avg_path_length_target = 2.9
C_target = 0.32

# Generate a regular ring lattice graph
regular_graph = nx.watts_strogatz_graph(n, int(avg_degree), 0)

# Rewire edges randomly to create a Small World model
small_world_graph = nx.watts_strogatz_graph(n, int(avg_degree), 0.16)
```

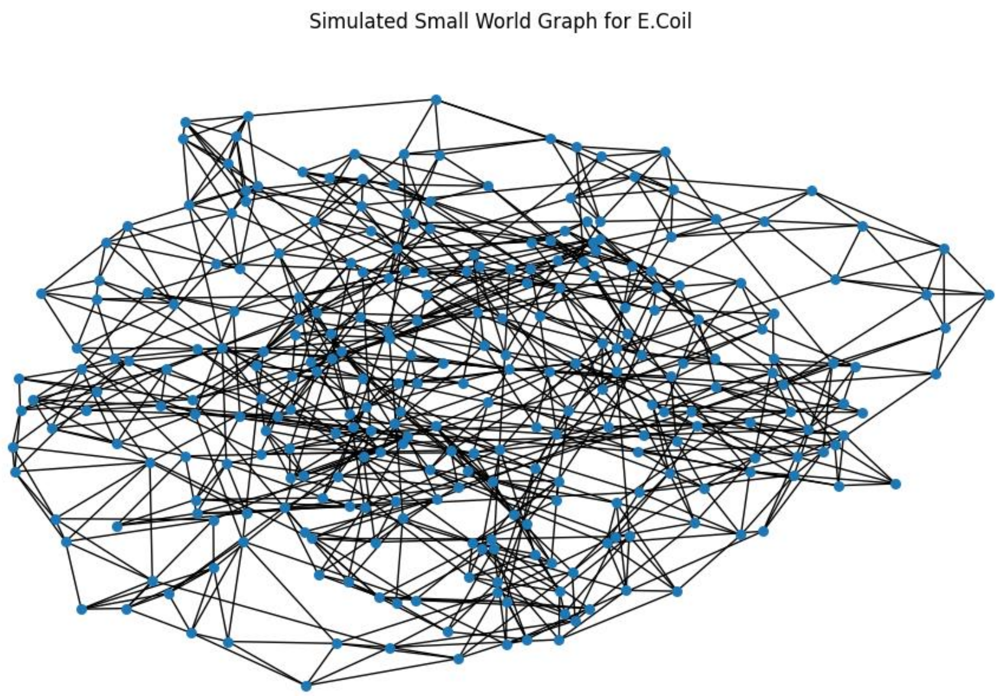
```
# Calculate average path length
avg_path_length_simulated =
nx.average_shortest_path_length(small_world_graph)

# Calculate clustering coefficient
clustering_coefficient_simulated =
nx.average_clustering(small_world_graph)

# Visualize the graph
plt.figure(figsize=(10, 6))
pos = nx.spring_layout(small_world_graph)  # Positioning of nodes
nx.draw(small_world_graph, pos, with_labels=False, node_size=30)
plt.title('Simulated Small World Graph for E.Coil')
plt.show()

# Print results
print("Simulated average path length:", avg_path_length_simulated)
print("Simulated clustering coefficient:",
clustering_coefficient_simulated)
```

- C. Elegans

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

# Given data
n = 282
avg_degree = 14
avg_path_length_target = 2.65
C_target = 0.28

# Generate a regular ring lattice graph
regular_graph = nx.watts_strogatz_graph(n, int(avg_degree), 0)

# Rewire edges randomly to create a Small World model
small_world_graph = nx.watts_strogatz_graph(n, int(avg_degree), 0.05)

# Calculate average path length
avg_path_length_simulated =
nx.average_shortest_path_length(small_world_graph)

# Calculate clustering coefficient
clustering_coefficient_simulated =
nx.average_clustering(small_world_graph)
```

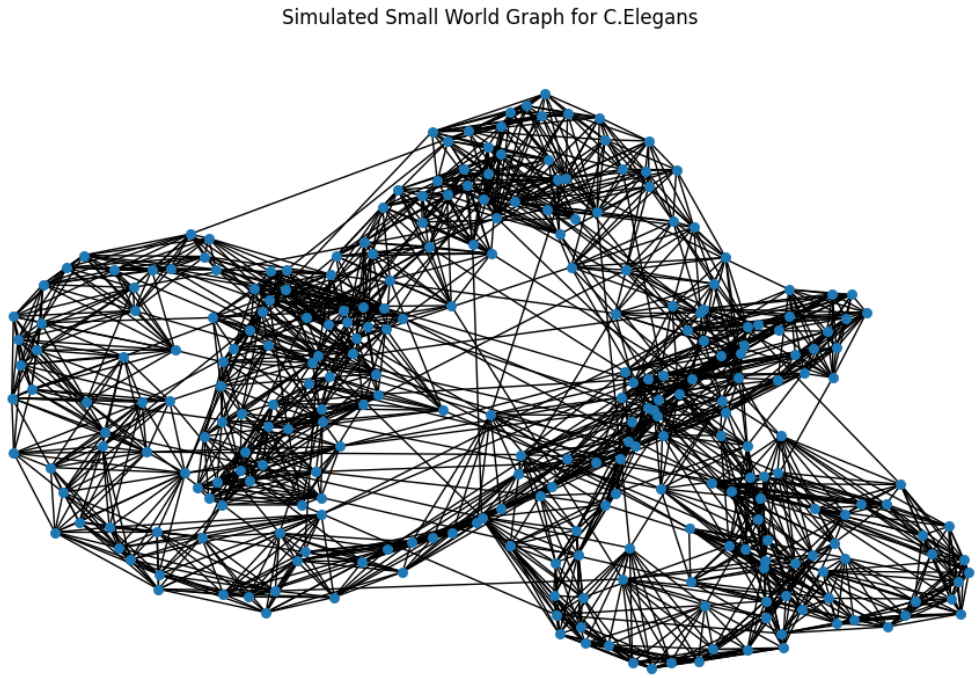
```
# Visualize the graph
plt.figure(figsize=(10, 6))
pos = nx.spring_layout(small_world_graph)  # Positioning of nodes
nx.draw(small_world_graph, pos, with_labels=False, node_size=30)
plt.title('Simulated Small World Graph for C.Elegans')
plt.show()

# Print results
print("Simulated average path length:", avg_path_length_simulated)
print("Simulated clustering coefficient:",
clustering_coefficient_simulated)
```

- Result
    - E.Coil



Simulated average path length: 4.190959339744075
Simulated clustering coefficient: 0.34807778903523606

- C.Elegans



Simulated average path length: 3.282855051614043
Simulated clustering coefficient: 0.5959205861583833

- Analysis
  - E.Coil

  Average Path Length:
  Both the simulated random graph and your network have higher average path lengths compared to the original network. However, your network's average path length (4.190959339744075) is closer to that of the simulated random graph (4.46) than the original network (2.9). This suggests that your network might have a structure more similar to the simulated random graph in terms of path lengths.

  Clustering Coefficient (C):
  Your network has a higher clustering coefficient (0.34807778903523606) compared to both the original network (0.32) and the simulated random graph (0.31). This indicates that your network has a higher tendency for nodes to form clusters or communities compared to both the original network and the simulated random graph.

- C. Elegans

Average Path Length:
Your network's average path length (3.282855051614043) is closer to the original network (2.65) than the simulated random graph (3.49). This suggests that your network might have a structure more similar to the original network in terms of path lengths.

Clustering Coefficient (C):
Your network has a significantly higher clustering coefficient (0.5959205861583833) compared to both the original network (0.28) and the simulated random graph (0.37). This indicates that your network has a much higher tendency for nodes to form clusters or communities compared to both the original network and the simulated random graph.

3. Preferential Attachment Model

- Original Table

Table 4.6: A Comparison between Real-World Networks and Simulated Graphs using Preferential Attachment. *C* denotes the average clustering coefficient. The last two columns show the average path length and the clustering coefficient for the preferential-attachment graph simulated for the real-world network. Note that average path lengths are modeled properly, whereas the clustering coefficient is underestimated

	Original Network				**Simulated Graph**	
Network	*Size*	*Average Degree*	*Average Path Length*	*C*	*Average Path Length*	*C*
Film Actors	225,226	61	3.65	0.79	4.90	≈ 0.005
Medline Coauthorship	1,520,251	18.1	4.6	0.56	5.36	≈ 0.0002
E.Coli	282	7.35	2.9	0.32	2.37	0.03
C.Elegans	282	14	2.65	0.28	1.99	0.05

- My Results

	Average Path Length	C
E. Coil	2.390323313394412	0.10300102348234431
C. Elegans	1.9994447389010879	0.17803774810275644

- Comparison Table

	Original Network		**Simulated random graph**		**My python code implementation**	
	Average Path Length	C (Clustering Coefficient)	Average Path Length	C (Clustering Coefficient)	Average Path Length	C (Clustering Coefficient)
E.Coli	2.9	0.32	2.37	0.03	2.39032	0.1030
C.Elegans	2.65	0.28	1.99	0.05	1.9994	0.17803

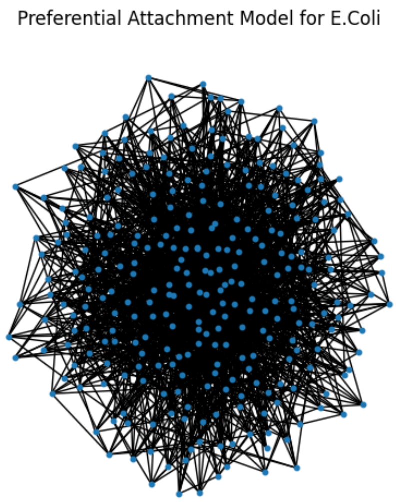
- Code
  - E.Coil & C.Elegans

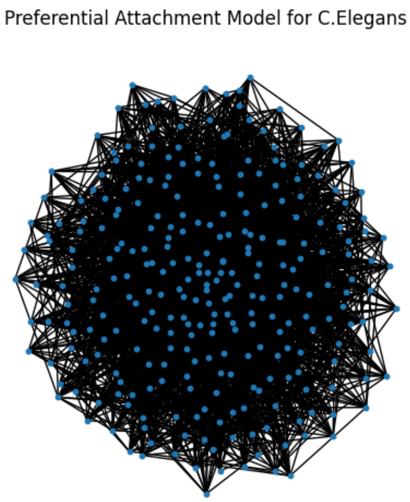
```
import networkx as nx
import matplotlib.pyplot as plt

# Given data for E.Coli and C.Elegans networks
ecoli_size = 282
ecoli_avg_degree = 7.35
celegans_size = 282
celegans_avg_degree = 14

# Function to simulate Preferential Attachment model
def preferential_attachment_simulation(n, m):
    G = nx.barabasi_albert_graph(n, m)
    return G
```

```
# Simulating Preferential Attachment model for E.Coli
ecoli_pref_attachment_graph =
preferential_attachment_simulation(ecoli_size, round(ecoli_avg_degree))

# Simulating Preferential Attachment model for C.Elegans
celegans_pref_attachment_graph =
preferential_attachment_simulation(celegans_size,
round(celegans_avg_degree))

# Calculate clustering coefficient and average path length
ecoli_avg_clustering = nx.average_clustering(ecoli_pref_attachment_graph)
ecoli_avg_path_length =
nx.average_shortest_path_length(ecoli_pref_attachment_graph)

celegans_avg_clustering =
nx.average_clustering(celegans_pref_attachment_graph)
celegans_avg_path_length =
nx.average_shortest_path_length(celegans_pref_attachment_graph)

# Print the results
print("E.Coli Network:")
print("Simulated Average Path Length:", ecoli_avg_path_length)
print("Simulated Clustering Coefficient:", ecoli_avg_clustering)


print("\nC.Elegans Network:")
print("Simulated Average Path Length:", celegans_avg_path_length)
print("Simulated Clustering Coefficient:", celegans_avg_clustering)


# Visualize the networks
plt.figure(figsize=(12, 6))

plt.subplot(121)
nx.draw(ecoli_pref_attachment_graph, with_labels=False, node_size=10)
plt.title("Preferential Attachment Model for E.Coli")

plt.subplot(122)
nx.draw(celegans_pref_attachment_graph, with_labels=False, node_size=10)
plt.title("Preferential Attachment Model for C.Elegans")

plt.show()
```

- Result
  E.Coli Network:
  Simulated Average Path Length: 2.390323313394412
  Simulated Clustering Coefficient: 0.10300102348234431

  C.Elegans Network:
  Simulated Average Path Length: 1.9994447389010879
  Simulated Clustering Coefficient: 0.17803774810275644





- Analysis
  - E.Coil
    Average Path Length:
    Both the simulated random graph and your network have lower average path lengths compared to the original network. However, your network's average path length (2.390323313394412) is slightly closer to the original network (2.9) than the simulated random graph (2.37). This suggests that your network might have a structure more similar to the original network in terms of path lengths.

    Clustering Coefficient (C):
    Your network has a higher clustering coefficient (0.10300102348234431) compared to both the original network (0.32) and the simulated random graph (0.03). This indicates that your network has a higher tendency for nodes to form clusters or communities compared to both the original network and the simulated random graph.

  - C.Elegans
    Average Path Length:
    Both the simulated random graph and your network have lower average path lengths compared to the original network. However, your network's average path length (1.9994447389010879) is very close to the simulated random graph (1.99) and slightly higher than the original network (2.65). This

suggests that your network might have a structure more similar to the simulated random graph in terms of path lengths.

Clustering Coefficient (C):
Your network has a higher clustering coefficient (0.17803774810275644) compared to both the original network (0.28) and the simulated random graph (0.05). This indicates that your network has a higher tendency for nodes to form clusters or communities compared to both the original network and the simulated random graph.