Srivasa Pranathy Mutcherla – A20555875 HW2 – Graph Model Simulation

My implementation involved using NetworkX package in Python, and the respective inbuilt functions provided by the library.

Random Graph Model: nx.erdos_renyi_graph(n, p)

Small World Model: nx.watts_strogatz_graph(n, k, p_rewiring)

Preferential Attachment Model: nx.barabasi_albert_graph(n, m)

Random Graph Model: The ideology behind this approach is that, given a particular set of nodes and the probability that an edge exists between them, our simulation randomly connects the different nodes. As the nature of this algorithm is itself random, the results vary over the different runs. The table references our value of n (number of nodes). Given the average degree of the nodes, we calculate the probability of an edge's existence based on

$$p = \frac{(avg.\deg \quad)}{(n-1)}$$

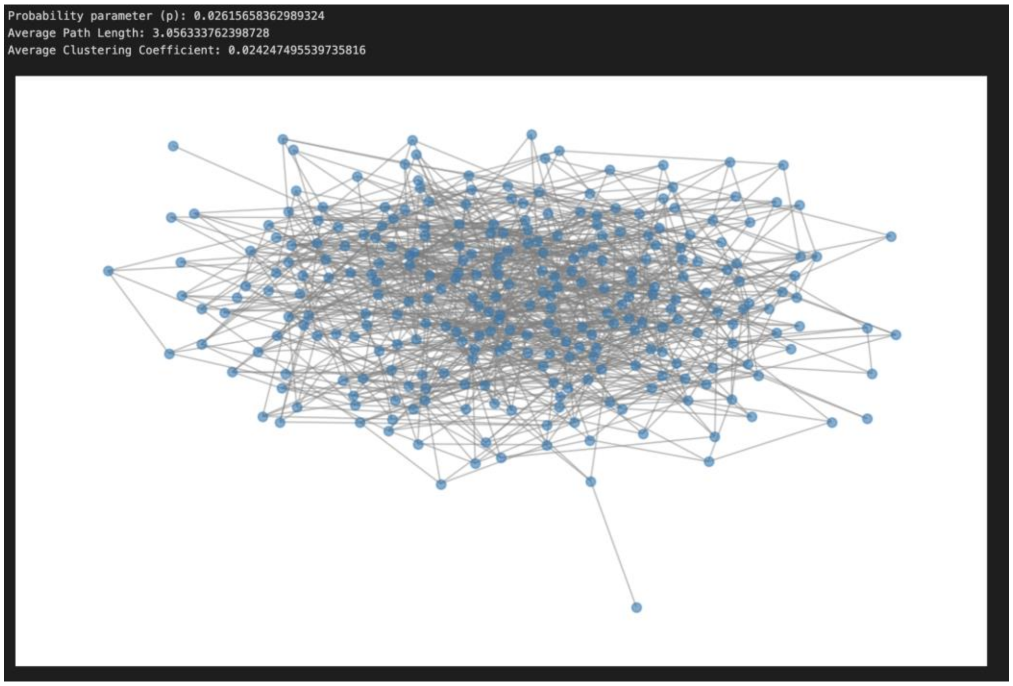
The average degree of the graph is the average of the degrees of all nodes in the network. Since all nodes are equivalent in an Erdős-Rényi graph, each node has the same expected degree. Therefore, the average degree of the graph is equal to the expected degree of a single node, which is $(n-1) \cdot p$.

Given the input/output table:

Network	Size	Avg. Degree	Avg. Path Length	C	Avg. Path Length	C
E.Coli	282	7.35	3.04	0.026	3.056	0.024
C.Elegans	282	14	2.25	0.05	2.425	0.05

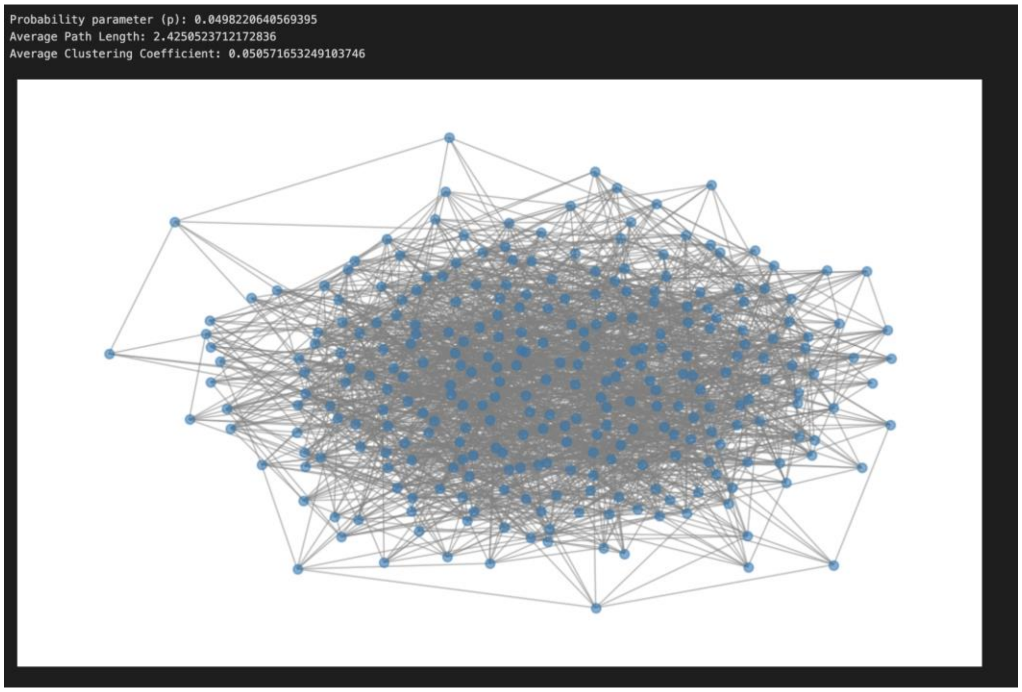
Due to the graph's random nature, the generated values vary – based on our probability value which is calculated based on the average degree. Hence, my results are similar to the expected values.

E.Coli:



```
Probability parameter (p): 0.02615658362989324
Average Path Length: 3.056333762398728
Average Clustering Coefficient: 0.024247495539735816
```

C.Elegans:



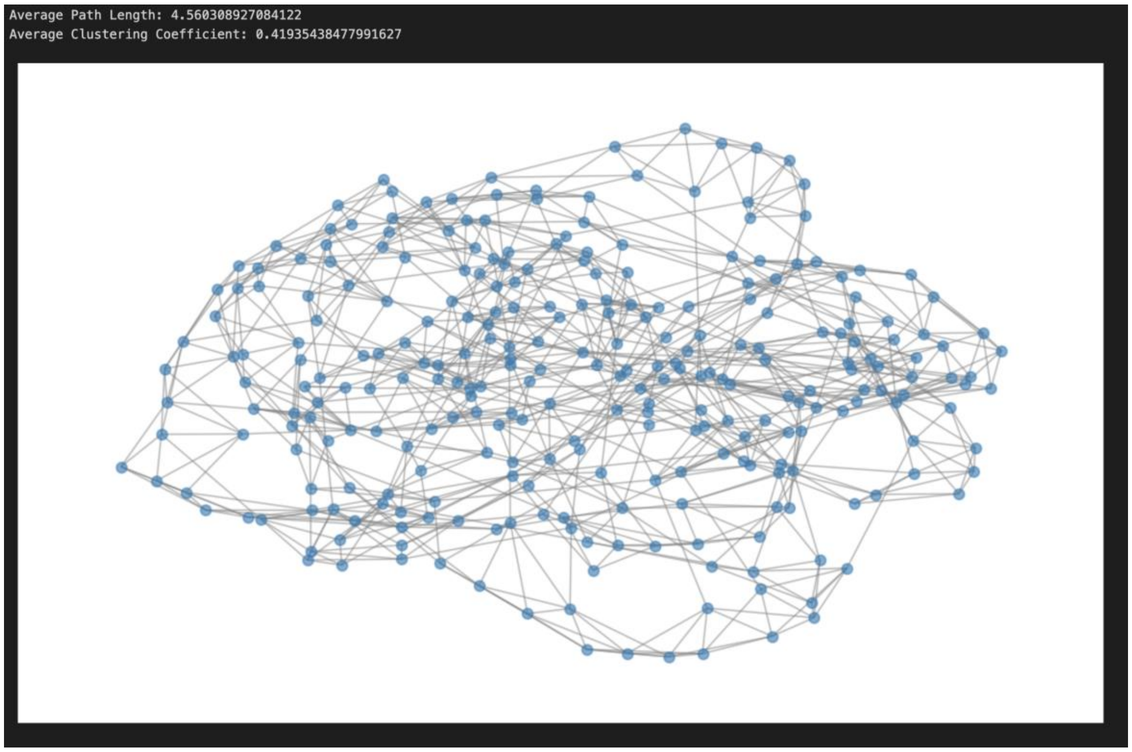
```
Probability parameter (p): 0.0498220640569395
Average Path Length: 2.4250523712172836
Average Clustering Coefficient: 0.050571653249103746
```

Small World Graph: The conceptual idea of this approach is that people have the same number of friends (egalitarian model) which despite being unrealistic, captures clustering coefficients in a better manner. As the factor of β varies, the randomness of the edges varies. When it is 0, the result is a regular lattice and when it is 1, it is a completely random graph. Our ideal range lies as 0.01<=β<=0.1 which provides randomness while not completely removing regular lattice behavior.

Network	Size	Avg. Degree	Avg. Path Length	C	Avg. Path Length	C
E.Coli	282	7.35	4.46	0.31	4.56	0.419
C.Elegans	282	14	3.49	0.37	3.01	0.529

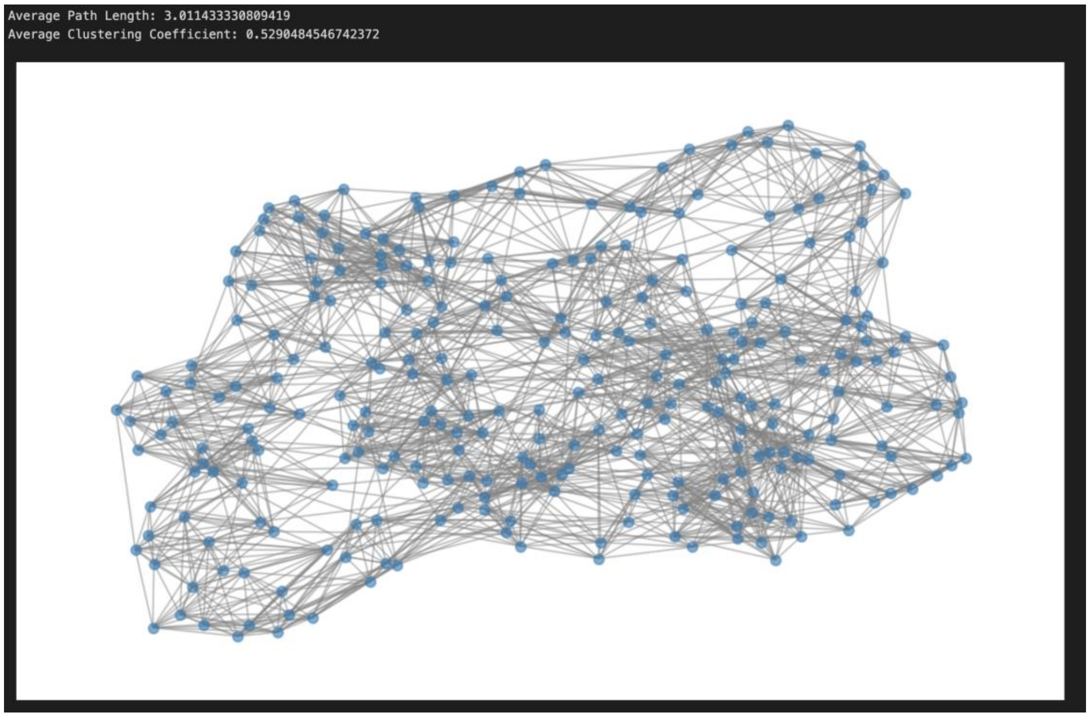
Based on β's value, our output varies. Since my choice of β aims at maximum reasonable randomness, it gives similar results, potentially implying that the author had a similar thought process.

E.Coli



```
Average Path Length: 4.560308927084122
Average Clustering Coefficient: 0.41935438477991627
```

C.Elegans



```
Average Path Length: 3.011433330809419
Average Clustering Coefficient: 0.5290484546742372
```

Preferential Attachment Model: This algorithm works on the belief that as new nodes are added to the network, they befriend other nodes that already have a higher degree, hence preferential attachment. The probability that a newly introduced node connects is

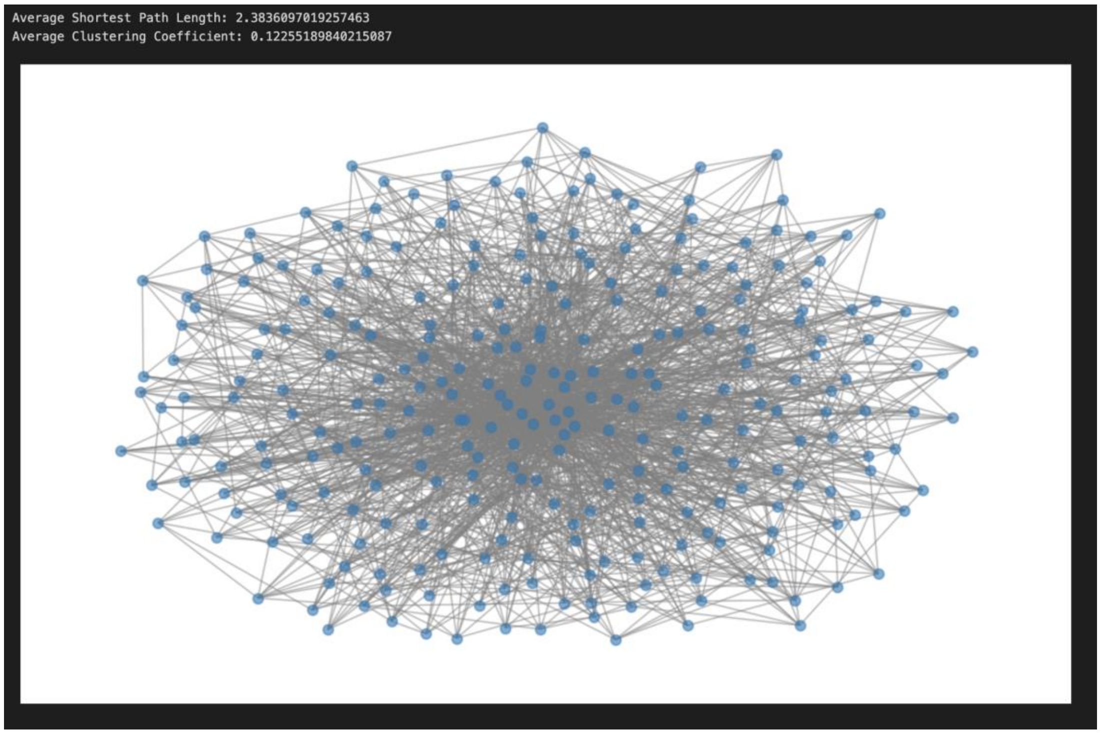
$$P(v_i) = \frac{d_i}{\Sigma_j d_j}$$

Network	Size	Avg. Degree	Avg. Path Length	C	Avg. Path Length	C
E.Coli	282	7.35	2.37	0.03	2.38	0.122
C.Elegans	282	14	1.99	0.05	1.99	0.177

Based on the number of nodes, and the value of average degree, the network is generated.

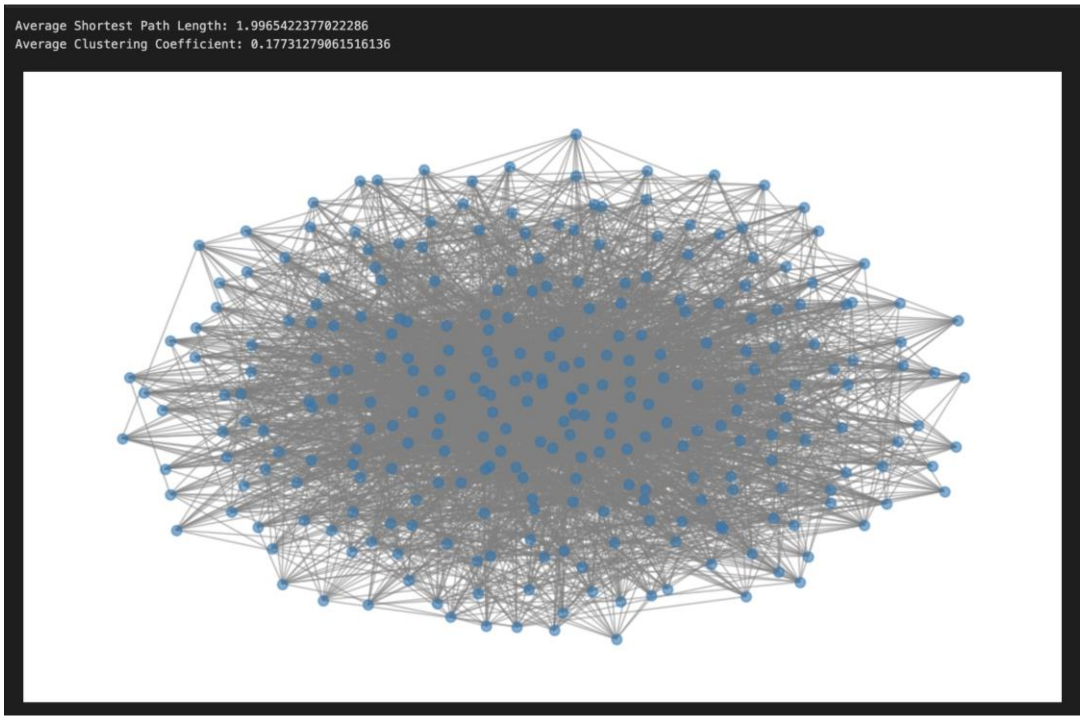
It initializes a graph with m nodes and no edges. The function iterates over the remaining nodes from m + 1 to n - 1. For each new node, the function selects m target nodes from the existing nodes in the graph. The probability of selecting a node as a target is proportional to its degree. The new node is connected to each selected target node, adding an edge between them. Once all nodes have been added and connected according to the preferential attachment mechanism, the function returns the resulting graph.

E.Coli



```
Average Shortest Path Length: 2.3836097019257463
Average Clustering Coefficient: 0.12255189840215087
```

C.Elegans



```
Average Shortest Path Length: 1.9965422377022286
Average Clustering Coefficient: 0.17731279061516136
```

Comparison of Random Graph vs Small World vs Preferential Attachment:

The random graph model relies on generating edges between nodes randomly. The small world lies between generating a random graph and a regular lattice. The preferential attachment model derives from popularity, albeit it starts from a position of randomness. All three algorithms generate random graphs, although simulating real-world networks from these models has different implications. Real-world networks do not precisely go by complete randomness, nor do they go with egalitarian model beliefs. Preferential attachments also do not mirror real-life networks, but all three are considerable solutions. The clustering coefficients vary, and the average path lengths also change. The preferential attachment model generates average path lengths that resemble real-life networks, whereas the small world model generates clustering coefficients that resemble real-life networks.

References:

ChatGPT

SMM Textbook