DS210 Final Project

Dataset: https://nijianmo.github.io/amazon/index.html
I am interested in investigating the relationship and structure of video game review data. By identifying whether there are clear clusters or communities within this network, I gain insight into customers preferences, which can lead to new market strategies for targeting the community who buys games. To perform this I will be performing a six degrees of separation by using breadth-first search and shortest path. The vertices (nodes) represents the reviewersID and games (ASIN), the edges represents the interactions between the entities. For example, if reviewer "A1" has reviewed "B1", then there is an edge between vertex A1 and B1. This graph will be an indirect graph as there is no direction, but if there is an edge from a reviewer to a game, it implies a mutual connection.

Within my code, I will be parsing the dataset containing the reviews and entry links to a reviewer ID and game ID. By using this data, we construct a graph where the vertices represent both the reviewers and the games, while the edges represent the relationship between them. To execute this, I will be using a "HashMaps" to represents the adjacency list fo the graph, where each key-value pair consists of a vertex and its corresponding connected vertices. After parsing the data, we will create a graph then utlize algorithms Breadth First Search to analyze the graph properties. I will be computing the average shortest path between vertices and through utlizing BFS for each vertex to calculate shortest path to other vertices. As well as implementing a function to test if the graph follows the "six degrees of separation" principle, which asserts that vertices should be connected by six or fewer edges. Finally, I will write test code to validate our graph implementations and ensure it behaves as expected.

Some difficulties I encountered is the size of the network, as it occupied a lot of memory usage and the time complexity of BFS (O(V+E)) led to long execution time. Overall, analyzing the whole dataset was inefficient as storing the graph structure can consume lots of memory and increase the time complexities due to its size. As a result, I had to use random sampling and reduce the number of vertices within my code to decrease execution time. However, this can lead to a loss of network and the original structure, which may not represent the original graph's pattern accurately. This could potentially skew the outputs leading to possible incorrect conclusions.

Output:

```
Graph 1: Average Shortest Path Length = 6.44
Six degrees of separation do not hold true for Graph 1.
Graph 2: Average Shortest Path Length = 6.91
Six degrees of separation do not hold true for Graph 2.
Graph 1 has a shorter average shortest path.
Time elapsed is: 306.593683s
```

I sampled 1500 unique reviewer IDs within my code to calculate the average shortest path and created two graphs to compare the two sampling to see if I received different results. The average shortest path indicates how many steps on average it took to travel between any two nodes (reviewer IDs or ASINs) in the graph. Then I checked if the shortest path length is less than 6 to determine if they conform to the six

degrees of separation theory, which neither of the graphs passed. As a result this means that it takes more than six steps to connect any two entities, meaning that the relationship are not particularly interconnected meaning the communicated shared little interest and products in similar games. Both graphs have similar average shortest path length which could indicate that both samples are relatively similar; if the data was performed utlizing the whole data, there is a possibility that it also did not pass the six degrees of separation. Due to random sampling, the graphs does not meet the six degrees of separation due to random variation, different types of games, and the geographical/demographic characteristic of the reviewers. I also included the computation time to highlight the need for optimization and more efficient algorithms and that if utlizing the whole dataset it could lead to an extremely long computation time.