Algorithms for Data Science
Assignment 1
Small World Phenomenon

The small world phenomenon refers to the proposition that we are all linked by short chains of acquaintances. The expression "six degrees of separation" refers to the idea that all pairs of people are at most six social connections away from each other.

In this assignment, you will experimentally verify if this property holds on a subset of Facebook data. You will be given a data set representing a set of people and acquaintances. Each entry stores the IDs of two people that know each other. For example

```
2 3
3 4
1 2
4 0
…
```

You should interpret the set of entries as the edges E of an undirected graph G = (V, E) where V is implicitly given by the IDs appearing in E. To complete this assignment, you should do the following:

1. Put all your code in a file called swp.py.

2. Write a function loadGraph(edgeFilename) that reads in the file of edge data as described above. Note that edgeFilename is a string storing the name of the edge data file. You will be provided with actual Facebook data to load into your program. This function should return an adjacency list representation of the corresponding undirected graph. A list of vertex IDs is not explicitly given but instead can be inferred from the edge data.

3. Implement a queue class, called MyQueue. Your class should include a constructor for creating an empty queue, as well as the standard methods enqueue, dequeue, and empty. It would also be useful to have a str method so you can use print to display the contents of the queue for debugging.

4. Implement a function BFS(G, s) that runs the breadth-first search (BFS) algorithm outlined in the course video/PowerPoint slides. It should run a BFS starting with source vertex s ∈ V. You should use your queue class implementation in the implementation of this function. It should return a list that contains the distance from s to every other vertex, v, in the graph. That is, the distance to Vertex 5 would be stored in Slot 5 of the

list. The graph will be passed using the adjacency list representation from Step 2 above.

5. Write a function distanceDistribution(G) to compute the distribution of all distances in G. Your function should return a dictionary that maps positive distances to frequency of occurrence. Specifically, the frequencies should be stored in percentage form. That is, 23.6% of all distances are three apart. Note that this might take a few minutes to run, so you might want to print out values every once in a while to show progress.

6. Write testing code at the bottom of your swp.py file to run the problem on the data file provided, printing out the final distribution dictionary.

7. To what extent does this network satisfy the small world phenomenon? Please put a comment section at the bottom of your code that answers this question.